COMPETITIVE EQUILIBRIUM APPROACH FOR LOAD BALANCING A COMPUTATIONAL GRID WITH COMMUNICATION DELAYS

K SHAHU CHATRAPATI[#], J UJWALA REKHA[*], DR. A. VINAYA BABU[*]

[#]Department of Computer Science and Engineering, JNTUH College of Engineering, Jagitial

[*]Department of Computer Science and Engineering, JNTUH College of Engineering, Hyderabad

ABSTRACT

Computational grids interconnect hundreds of heterogeneous computing resources from geographically remote sites, designed to meet the large demands of many users from scientific and business domains. A job initiated at one site can be executed by any of the computing resources. Therefore, response time of a job includes processing delay at the site of execution and communication delay for transferring the job from the site of initiation to the site of execution. Load balancing is allocation of jobs to available resources so as to optimize a given objective function. The objective can be achieving a system optimal solution, which tries to minimize the mean response time of all users or an individual optimal solution which tries to minimize each user's response time. Previous works on load balancing either considered only system optimal objective or individual optimal objective. This paper introduces competitive equilibrium solution, a pricing mechanism for load balancing that independently and simultaneously achieves both system optimal objective and individual optimal objective.

KEYWORDS: *Computational Grid, Competitive Equilibrium, Nash Equilibrium, Cooperative Load Balancing, Non Cooperative Load Balancing*

1. INTRODUCTION

Computational grid is a form of distributed system, where a large number of computing resources are connected by a network to solve science, engineering, and business problems that require a great number of computer processing cycles. A job or an application usually requires the resources from more than one owner. So, the grid computing system should be able to assign the jobs from various users to the different computing resources efficiently, and utilize the resources of unused devices [13]. The main goal of the load balancing is thus, to efficiently and fairly distribute load across the resources, so as to achieve individual optimality and system optimality.

From the perspective of optimization, load balancing can be formulated as either non-cooperative load balancing ([15],[9],[3],[11]) or as cooperative load balancing ([1],[2]). The game theoretic approach to non-cooperative load balancing using Nash equilibrium solution considers individual response times as their objective and do not consider mean response time of all users. On the other hand, cooperative load balancing considers optimization of the entire system, i.e., optimization of mean response time of all users and do not take into account each user's individual objective. Competitive equilibrium approach for load balancing is considered in [14] which achieves both system optimality and individual optimality; however, it does not take into account communication delay. This study investigates competitive equilibrium load balancing taking into account communication delay. Competitive equilibrium is a pricing mechanism that simultaneously and independently optimizes both system objective and each user's individual objective.

In the capitalist economy, crucial regulatory functions such as ensuring stability, efficiency, and fairness are relegated to pricing mechanisms. Thus, competitive equilibrium theory of equilibrium prices gained important place in mathematical economics. The study of competitive equilibrium theory was started by Walras [7], dating back to1870s. In the Walrasian model, each agent i has an initial endowment of divisible goods $\mathbf{w_i} = (\mathbf{w_{i1}}, .., \mathbf{w_{in}}) \in R_+^n$, and preferences for consuming goods described by utility function $u_i = R_+^n \rightarrow R_+$. At given prices, each agent *i* sells their entire initial endowment and then uses the revenue to buy a bundle of goods $\mathbf{x_i} = (\mathbf{x_{i1}},..,\mathbf{x_{in}}) \in R_+^n$, such that their utility $u_i(\mathbf{x})$ is maximized.

Walras posed a question whether prices could be determined for every good such that for each agent *i*, there is a bundle of goods such that their utility is maximized (individual optimality) and the market has neither shortage nor surplus (system optimality). In other words, an equilibrium is a set of prices $\mathbf{p} = (\mathbf{p_1},..,\mathbf{p_n}) \in R_+^n$, such that

(1) For each agent *i* there is a bundle of goods $\mathbf{x_i} = (\mathbf{x_{i1}},..,\mathbf{x_{in}}) \in R_+^n$, such that vector $\mathbf{x_i}$ is a maximizer of $u_i(\mathbf{x})$ subject to the constraints

$$\sum_{j=1}^{n} \mathbf{p_j} * \mathbf{x_{ij}} \leq \sum_{j=1}^{n} \mathbf{p_j} * \mathbf{w_{ij}} \qquad (1)$$

(2) And for each good j,

$$\sum_{i=1}^{n} \mathbf{x_{ij}} \leq \sum_{j=1}^{n} \mathbf{w_{ij}} \qquad (2)$$

Arrow and Debreu in 1954 [5], jointly showed that such an equilibrium would exist under very mild conditions if the utility functions are concave by applying Kakutani's fixed point theorem.

Fisher [4], independently modeled the competitive equilibrium market in 1891. In Fisher's market model there are two kinds of agents: **m** buyers and **n** divisible goods. Each buyer *i*, has money $\mathbf{e_i}$ and each good *j* has an initial endowment $\mathbf{b_j}$ of the good. Utility function for consuming goods is given by $u_i = R_+^n \rightarrow R_+$. Equilibrium prices is an assignment of prices $\mathbf{p} = (\mathbf{p_1},..,\mathbf{p_n})$ to goods, so that when every consumer *i*, buys an optimal bundle of goods $\mathbf{x_i} = (\mathbf{x_{i1}},..,\mathbf{x_{in}}) \in R_+^n$, then the market clears, i.e., all the money is spent and all the goods are sold. In other words, prices $\mathbf{p} = (\mathbf{p_1},..,\mathbf{p_n}) \in R_+^n$ are equilibrium prices if

(1) For each buyer *i*, there is a bundle of goods $\mathbf{x_i} = (\mathbf{x_{i1}},..,\mathbf{x_{in}}) \in R_+^n$ such that vector $\mathbf{x_i}$ is a maximizer of $u_i(\mathbf{x})$ and

$$\sum_{j=1}^{n} \mathbf{p_j} * \mathbf{x_{ij}} = \mathbf{e_i} \qquad (3)$$

(2) And for each good j,

$$\sum_{i=1}^{m} \mathbf{x_{ij}} = \mathbf{b_j} \qquad (4)$$

It can easily be observed that the Fisher's model is a special case of Walras model, when money is considered a good.

In this study, Fisher's market model is adopted where buyers are users, and goods are computing resources. The competitive equilibrium problem of load balancing is then finding equilibrium prices for the computing resources, and then determining allocation of user jobs to the resources at these prices, such that each user optimizes her objective function, subject to her budget constraints.

2. GRID SYSTEM MODEL

We consider a grid system of **n** heterogeneous nodes (computing resources) connected by a communication network shared by **m** users. The terminology and assumptions are similar to [12]. The job arrival rate of user *j* job at node *i* is $\boldsymbol{\varphi_i^j}$.

Total arrival rate of user *j* jobs is $\boldsymbol{\varphi^j} = \sum_{k=1}^{n} \boldsymbol{\varphi_k^j}$.

All the jobs in the system are assumed to be of same size. The service rate of node *i* is $\boldsymbol{\mu_i}$.

Out of user *k* jobs arriving at node *i*, the ratio $\mathbf{x_{ij}^k}$ of jobs is forwarded upon arrival through the communication means to another node $(j \neq i)$ to be processed there. The remaining ratio $\mathbf{x_{ii}^k} = 1 - \sum_{j \neq i} \mathbf{x_{ij}^k}$ is processed at node *i*.

That is, the rate $\boldsymbol{\varphi_i^k} \mathbf{x_{ij}^k}$ of user *k* jobs that arrive at node *i* are forwarded through the communication means to node *j*, while the rate $\boldsymbol{\varphi_i^k} \mathbf{x_{ii}^k}$ of user *k* jobs are processed at arrival node. Therefore, a set of values for $\mathbf{x_i^k}$ $(\mathrm{i} = 1,.., \mathrm{n}; \mathrm{k} = 1,.., \mathrm{m})$ are to be chosen where

$$x_i^k = \left(x_{i1}^k, \ldots, x_{in}^k\right) \quad (5)$$

is an n-dimensional vector such that

$$\sum_{j=1}^{n} x_{ij}^k = 1 \quad \text{for all } i = 1,..,n \quad (6)$$

$$x_{ij}^k \geq 0 \quad \text{for all } i = 1,..n;$$

$$j = 1,..,n;\ k = 1,..,m \quad (7)$$

and $$\sum_{k=1}^{m} \sum_{j=1}^{n} \varphi_j^k * x_{ji}^k < \mu_i$$

$$\text{for all } i = 1,..n \quad (8)$$

An nn-dimensional vector x^k is called the strategy profile of user *k* where

$$x^k = \left(x_1^k, \ldots, x_n^k\right) \quad (9)$$

An nnm-dimensional vector **x** is called the global strategy profile where

$$\mathbf{x} = \left(x_1^1, \ldots, x_n^1; x_1^2, \ldots, x_n^2; \ldots\ldots; x_1^m, \ldots, x_n^m\right)$$

or (10)

$$\mathbf{x} = \left(x^1, \ldots, x^m\right)$$

If each node is modeled as an M/M/1 queuing system [6], then the expected node delay at node *i* is as follows

$$F_i(\mathbf{x}) = \frac{1}{\mu_i - \beta_i} \quad (11)$$

where β_i is the load on node *i* and given as

$$\beta_i = \sum_{k=1}^{m} \sum_{j=1}^{n} \varphi_j^k * x_{ji}^k \quad (12)$$

Clearly, $F_i(\mathbf{x})$ is a strictly increasing, convex, and continuously differentiable function of x^j (j = 1,…,m).

We assume as in [12], that the expected communication delay of forwarding user *k* jobs at node *i* to node *j* is independent of two nodes but dependent on the total traffic through the network. Examples of such a case are local area networks and satellite communication systems, where the communication delay between any two nodes (or stations), depends on the total traffic generated by all the nodes (or stations).

In our grid system model, the total traffic through the network is denoted by λ, where

$$\lambda = \sum_{j=1}^{m} \lambda^j \quad (13)$$

and λ^j is the traffic through the network due to user *j* jobs given as follows

$$\lambda^j = \frac{1}{2} \sum_{i=1}^{n} \left|\varphi_i^j - \beta_i^j\right| \quad (14)$$

where β_i^j is the contribution on the load of node *i* by user *j* jobs given as

$$\beta_i^j = \sum_{k=1}^{n} \varphi_k^j x_{ki}^j \quad (15)$$

If the communication network is modeled as an M/M/1 queuing system [6], the expected communication delay of any job is given as

$$G(\lambda) = \frac{t}{\left(1 - t \sum_{k=1}^{m} \lambda^k\right)} \quad (16)$$

where *t* is the mean communication time for sending and receiving a job from one node to the other for any user. Clearly, $G(\lambda)$ is a positive, non-decreasing, convex, and continuously differentiable function of λ.

The overall response time of user *j* job is the sum of expected node delay at each node *i,* and expected communication delay given as follows

$$T^j(\mathbf{x}) = \frac{1}{\varphi^j} \sum_{i=1}^{n} \beta_i^j F_i(\mathbf{x}) + \frac{\lambda^j}{\varphi^j} G(\lambda) \quad (17)$$

The mean response time of all jobs is given by

$$T(\mathbf{x}) = \frac{1}{\varphi} \sum_{j=1}^{m} \varphi^j T^j(\mathbf{x}) \quad (18)$$

The best response time for user *j* job is a solution to the following optimization problem

$$\min_{\mathbf{x}^{\mathbf{j}}} T^{j}(\mathbf{x}) \qquad (19)$$

subject to the constraints (6) to (8).

3. COMPETITIVE EQUILIBRIUM LOAD BALANCING

At first the grid system model described in the previous section is translated to Fisher's market model, where buyers are users and goods are computing resources. Each user *j* (j = 1,…,m) is endowed a "monetary" budget $\mathbf{w_i} \geq 0$ and use it to purchase computing resources. However, $\mathbf{w_i}$ does not represent real money, but artificial and can be interpreted as "importance weight". If $\mathbf{w_i} = 1$ for all users, then all users are treated uniformly important. Each user *j* has utility function $u^{j}(\mathbf{x}) = -T^{j}(\mathbf{x})$, to denote her preferences for different bundles of goods. The price for executing unit job at node *i* is $\mathbf{p_i}$, where $\mathbf{p_i}$ like $\mathbf{w_i}$ is not real money but artificial, which is used to denote "ranking" of computing resources.

The competitive equilibrium problem of load balancing is to find a set of prices and allocation of jobs to computing resources such that each user maximizes her utility, subject to her budget constraints, and the market clears (i.e., all money is spent) .

It can be stated formally, as determining prices $\mathbf{p} = (\mathbf{p_1},..,\mathbf{p_n})$ and load fractions $\mathbf{x} = (\mathbf{x}^{\mathbf{1}},...,\mathbf{x}^{\mathbf{m}})$ such that **X** is a maximizer of

$$\max_{\mathbf{x}^{\mathbf{j}}} u^{j}(\mathbf{x}) \quad \text{for all } j = 1,..,m \qquad (20)$$

subject to the constraints (6) to (8) and market clearing condition given by

$$\sum_{i=1}^{n} \mathbf{p_i} * \boldsymbol{\beta}_{\mathbf{i}}^{\mathbf{j}} = \mathbf{w_j} \quad \text{for all } j = 1,..,m \qquad (21)$$

where $u^{j}(\mathbf{x})$ is strictly continous, concave, and continuously differentiable function of $\mathbf{x}^{\mathbf{j}}$ $(j = 1,..,m)$. Also $\mathbf{x}^{\mathbf{j}} \subseteq R_{+}^{n}$ and a closed convex set, bounded from below. According to the lemma of abstract economy developed by Debreu [4], the necessary and sufficient conditions for the existence of competitive equilibrium are satisfied, hence there exists a competitive equilibrium for the given load balancing problem.

The competitive equilibrium for load balancing is computed by price adjustment process called tâtonment trail and error introduced by walras [7].The users take the prices as given and determine their load fractions at each node. The price of each node is adjusted in proportion to the magnitude of aggregate load due to all users at that node. From the law of supply and demand, the price for executing a job at a node is increased if the demand (aggregate load due to all users) is more, and price for executing a job at a node is decreased if the demand is less. In each iteration, the users recalculate their loads at each node, upon receiving the newly adjusted prices, and in response to the newly calculated loads, the prices are adjusted. The process is continued until prices converge to equilibrium.

This is an artificial trade, where price **p** and budget **w** do not have any physical interpretations and have no outside use. They are only an economic means for achieving individual and system optimality. The meaningful output of our problem is only the load distribution.

We present below the algorithm for computing competitive equilibrium solution (CES) for the load balancing problem.

3.1. Algorithm (CES)

Input

Node Processing Rates:

$$\boldsymbol{\mu}_{\mathbf{1}},..., \boldsymbol{\mu}_{\mathbf{n}}$$

Job Arrival Rates:

$$\boldsymbol{\varphi}_{\mathbf{i}}^{\mathbf{j}} \left(\text{for all } j = 1,..,m; i = 1,...,n\right)$$

Output

Load Fractions $\mathbf{x}^{\mathbf{1}},...,\mathbf{x}^{\mathbf{m}}$

1. Initialization

 1.1. $\mathbf{w_j} \rightarrow 1$ for all j = 1,..,m

 1.2. $\mathbf{p_i} \rightarrow 1/n$ for all i = 1,..,n

2. Loop

 2.1. At prices $\mathbf{p_1},..,\mathbf{p_n}$ compute $\mathbf{x}^{\mathbf{1}},...,\mathbf{x}^{\mathbf{m}}$ such that each user maximizes her utility function (20) subject to the constraints (6) to (8).

 2.2. Obtain market clearing error, α given as follows

$$\alpha = \sqrt{\sum_{j=1}^{m} \xi_j^2} \qquad (22)$$

where ξ_j is given by

$$\xi_j = w_j - \sum_{i=1}^{n} p_i * \beta_i^j \qquad (23)$$

2.3. Adjust the prices $p_1,..,p_n$ in proportion to aggregate demands

Until α ≤ error tolerance

4. NUMERICAL SIMULATIONS

A computer model is run to evaluate the proposed scheme (CES) and two other schemes - Nash equilibrium solution (NES) and global optimal solution (GOS). The performance metrics used are the mean response time of all user jobs, individual response time of each user job and fairness index. The fairness index is the measure of fairness of allocation of resources to the users and is given as follows

$$FI = \frac{\left(\sum_{i=1}^{m} T^j(\mathbf{x})\right)^2}{n * \sum_{i=1}^{m} \left(T^j(\mathbf{x})\right)^2} \qquad (24)$$

If FI=1, the system is 100% fair to all users. FI decreases when, differences on $T^j(\mathbf{x})$ increases and the load balancing scheme favors only few users.

The other two schemes are described below-

i. Global Optimal Solution (GOS) –

In this, the expected mean response time of all user jobs is minimized. The loads β_i^j (for all j = 1,.., m; i = 1,..., n) are obtained by solving the following optimization problem

$$\min_{\mathbf{x}} T(\mathbf{x}) \qquad (25)$$

subject to the constraints (6) to (8).

ii. Nash Equilibrium Solution (NES) –

The loads β_i^j (for all j = 1,.., m; i = 1,..., n) for Nash equilibrium solution are obtained by solving the optimization problem given by (19) (for all j = 1,..,m), subject to the constraints (6) to (8).

Nash equilibrium solution is obtained by first, initializing strategy $\mathbf{x}^i$ of each user i to zero vector. Then each player updates its strategy $\mathbf{x}^i$ by solving the optimization problem (19) one after the other. Nash equilibrium is reached when no player can change its strategy $\mathbf{x}^{i*}$ and decrease its response time by choosing a different strategy $\mathbf{x}^{i*'}$ when the other user's strategies are fixed.

4.1. Results

The three solutions are evaluated under various loads and configurations to study the impact of system utilization and heterogeneity, on each user's individual response time, mean response time of all user jobs, and fairness index of the system.

4.1.1. Effect of System Utilization

System utilization (ρ) is the ratio of the total arrival rate of the system, to the aggregate service rate of the system, as given below:

$$\rho = \frac{\varphi}{\sum_{i=1}^{n} \mu_i} \qquad (26)$$

A heterogeneous model of 16 computers with four different service rates shared by 10 users is considered. The system configuration of the computers is given in Table 1. For a given system utilization, total job arrival rate φ is obtained from (26) above. From the φ obtained, the job arrival rate of user *j* job (φ^j) is determined from the total job arrival rate φ as $\varphi^j = \varphi * q^j$, where q^j, the job arrival fraction of user *j* is given in Table 2.

Table 1. System Configuration

Number of Computers	5	5	4	2
Service Rate (jobs/sec)	10	20	50	100

The job arrival rate of user *j* jobs to each computer *i*, (i = 1,..,n) is determined as $\varphi_i^j = \varphi^j * q_i$ where the fractions q_i are given in Table 3. The mean communication time *t* is taken to be 0.01sec.

Table 2. Job arrival fractions q^j of each user

User	1	2	3-6	7-9	10
q^j	0.3	0.2	0.1	0.01	0.07

Table 3. Job arrival fractions q_i to each computer

No. of Computers	1-2	3-6	7-11	12-14	15-16
q_i	0.01	0.02	0.04	0.1	0.2

Figures 1 and 2 present the mean response time of all users, and the fairness index of the system respectively for values of system utilization ranging from 10% to 90%. In all three schemes, while mean response time is increasing with increasing system utilization, fairness index is decreasing with increasing system utilization. Also, it can be seen that expected mean response time is better in GOS than in NES and CES. However, fairness index of GOS is lesser than both NES and CES. And we observe that the mean response time of all users in CES is close to GOS and is fairer than both NES and GOS.

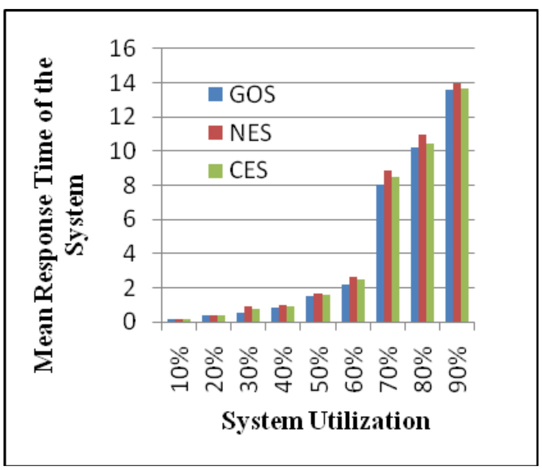

Figure 1. System Utilization Vs Mean Response Time of all User Jobs

Figures 3, 4, and 5 present the individual response times of each user at system utilizations of 10%, 50%, and 90% respectively. We observe that in most of the cases CES performs better than GOS and NES. Therefore individual optimality of CES is better than NES.

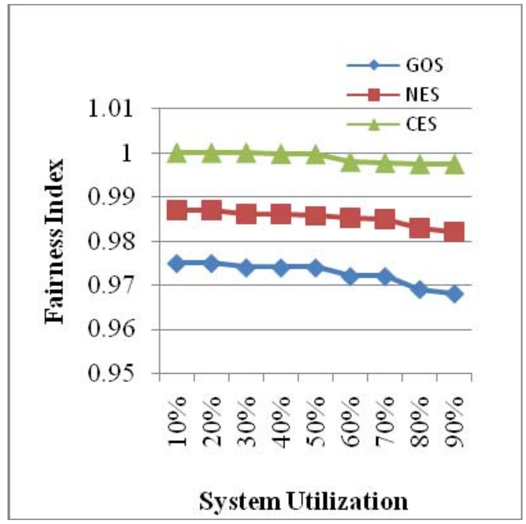

Figure 2. System Utilization Vs Fairness Index

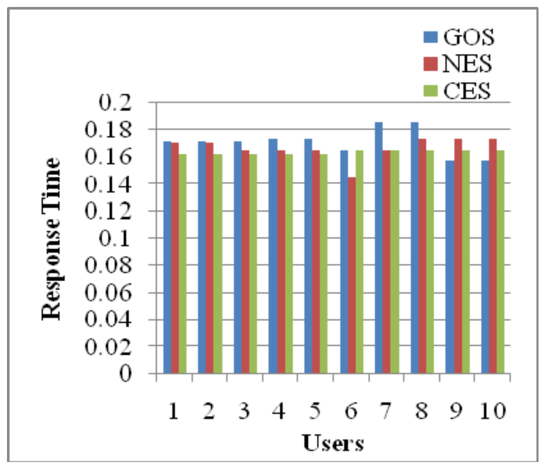

Figure 3. Response Time of Each User at System Utilization of 10%

4.1.2. Effect of Heterogeneity

Heterogeneity can be measured in terms of speed skewness, which is the ratio of maximum processing rate to minimum processing rate of the grid computers. The impact of heterogeneity on mean response time of the system and fairness index is investigated by varying speed skewness from 2 to 12 as given in Table 4 and presented in figures 6 and 7 respectively for system utilization of 50%.

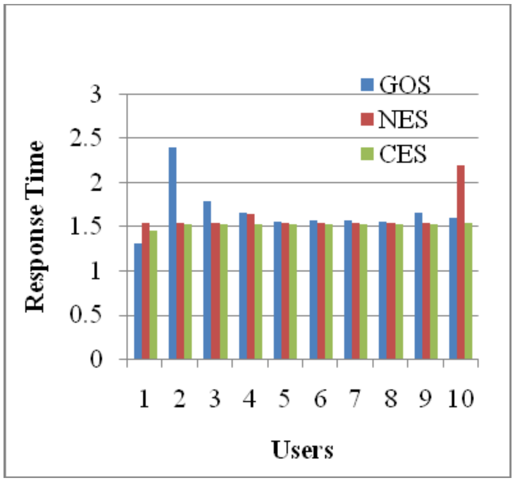

Figure 4. Response Time of Each User at System Utilization of 50%

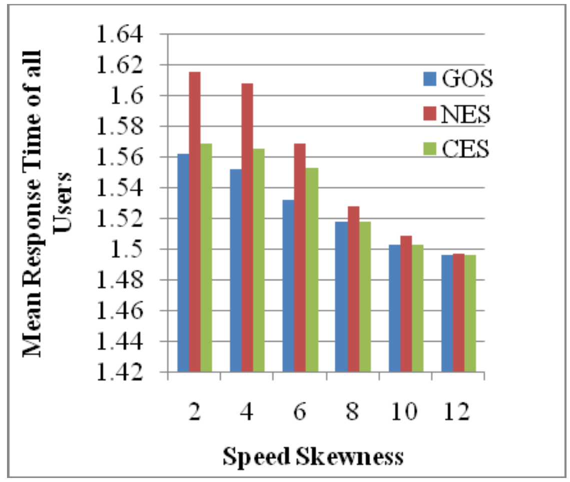

Figure 6. Heterogeneity Vs Mean Response Time of all User Jobs

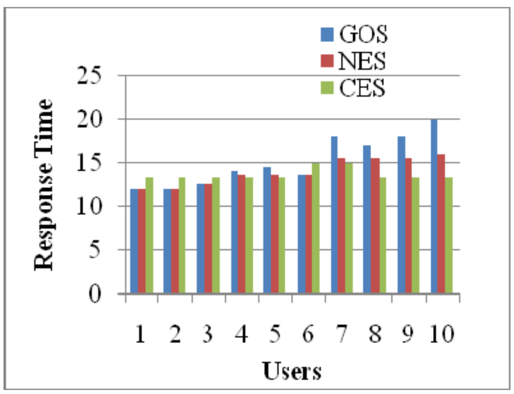

Figure 5. Response Time of Each User at System Utilization of 90%

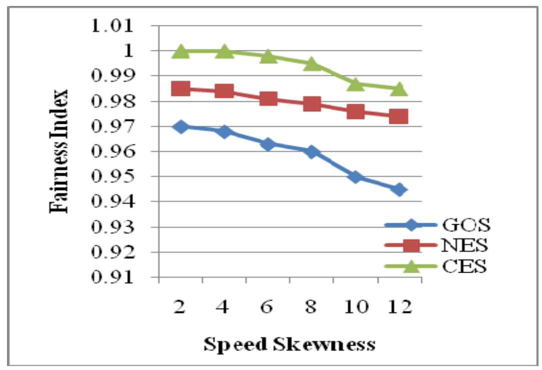

Figure 7. Heterogeneity Vs Fairness Index of the System

Table 4. System Parameters

Speed Skewness	2	4	6	8	10	12
μ_i of c_1, c_2	20	40	60	80	100	120
μ_i of c_3 to c_{16}	10	10	10	10	10	10

We observe that the mean response time of all users is better in GOS than in NES and CES, and mean response time of all users in CES and NES is almost the same as in GOS with increasing speed skewness.

Moreover fairness index in all the three schemes decreased with increasing speed skewness. However, it is greater in the case of CES. Therefore, CES simultaneously and individually achieved both system optimality and user-optimality.

5. CONCLUSIONS

Our study proposes competitive equilibrium solution for load balancing a computational grid considering communication delays.

A computer model of a grid is ran with various system loads and configurations and compared with two other schemes – global optimal solution and Nash equilibrium solution. Though global optimal solution achieved better mean response time, it is not fair to all users. On the other hand, Nash equilibrium solution achieved better fairness at the expense of increased mean response time. The mean response time in competitive equilibrium solution is close to global optimal solution and at the same time is fairer than Nash equilibrium solution. Therefore, competitive equilibrium solution achieved both system optimality and individual optimality simultaneously.

In our study, we considered static schemes for load balancing. In the future, the model can be extended to consider run time state information to make better load balancing decisions.

REFERENCES:

[1] Asser N. Tantawi, Don Towsley, Optimal Static Load Balancing in Distributed Computer Systems, *Journal of ACM,* 32(2), 445-465 (1985).

[2] Chonggum Kim, Kameda, H., An Algorithm for Optimal Static Load Balancing in Distributed Computer Systems, *IEEE Transactions on Computers,* 41(3), 381-384 (1992).

[3] Daniel Grosu, Anthony T. Chronopoulos, Noncooperative Load Balancing in Distributed Systems, *Journal of Paralled and Distributed Computing,* 65(9), 1022-1034 (2005).

[4] H. Scarf, The Computation of Economic Equilibria, *In Cowles Foundation Monograph No. 24,* Yale University Press, New Haven (1973).

[5] K.J. Arrow and G. Debreu, Existence of an Equilibrium for a Competitive Economy, *Econometrica,* 22(3), PP.265-290 (1954).

[6] Kleinrock, L., Queuing Systems-Theory, Volume 1, *John Wiley& Sons*

[7] L. Walras, *Elements d'Economie Politique Pure; OU, Theorie de la richesse sociale*. Elements of Pure Economics; or the theory of social wealth, Lausanna, Paris (1874).

[8] Ming-Hua Lin, Jung-Fa Tsai, Yinyu Ye, Budget Allocation in a Competitive Communication Spectrum Economy, *Eurasip Journal on Advances in Signal Processing* (2009).

[9] Preetam, Ghosh, Roy, Das, Basu, A Pricing Strategy for Job Allocation in Mobile Grids Using a Noncooperative Bargaining Theory Framework, *Journal of Parallel and Distributed Computing,*65(11), 1366-1383 (2005).

[10] R. Jain, D. Chiu, and W. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems, *DEC Research Report TR-301*(1984).

[11] Riky Subrata, Albert Y. Zomaya, Game Theoretic Approach for Load Balancing in Computational Grids, *IEEE Transactions on Parallel and Distributed Systems,* 19(1), 66-76 (2008).

[12] Satish Penmatsa, Anthony T. Chronopoulos, Price-based User-Optimal Job Allocation Scheme for Grid Systems, *In the Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium* (2006).

[13] Satish Penmatsa, Anthony T. Chronopoulos, Job Allocation Schemes in Computational Grids Based on Cost Optimization, *In the Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium* (2005).

[14] Shahu Chatrapati, K., Ujwala Rekha, J.,Vinaya Babu, A., Competitive Equilibrium Approach for Load Balancing in Computational Grid, *In the Proceedings of the international conference on Advances and Emerging Trends in Computing Technologies*, pp.63-68, (2010)

[15] Spata, M.O., A Nash Equilibrium Based Algorithm for Scheduling Jobs on a Grid Cluster, *In the proceedings of 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises,* 18(20), 251-252 (2007)