Chapter 5

Competitive Equilibrium Approach for Load Balancing a Computational Grid with Communication Delays

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 ([3],[15]), which tries to minimize the mean response time of all users or an individual optimal solution ([17],[68],[71],[85]) which tries to minimize each user’s response time. Previous works on load balancing either considered only system optimal objective or individual optimal objective. Competitive equilibrium approach for load balancing is considered in [46] 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 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.

5.1. 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 [81]. The job arrival rate of user \( j \) job at node \( i \) is \( \phi_{ji} \). Total arrival rate of user \( j \) jobs is \( \phi^j = \sum_{k=1}^{n} \phi_{kj} \). All the jobs in the system are assumed to be of same size. The service rate of node \( i \) is \( \mu_i \).

Out of user \( k \) jobs arriving at node \( i \), the ratio \( x_{ij}^k \) of jobs is forwarded upon arrival through the communication means to another node \(( j \neq i )\) to be processed there. The remaining ratio \( x_{ii}^k = 1 - \sum_{j \neq i} x_{ij}^k \) is processed at node \( i \).
That is, the rate $\phi_{ij}^k$ of user $k$ jobs that arrive at node $i$ are forwarded through the communication means to node $j$, while the rate $\phi_{ii}^k$ of user $k$ jobs are processed at arrival node. Therefore, a set of values for $x_{ij}^k (i = 1,..,n; k = 1,..,m)$ are to be chosen where

$$x_i^k = (x_{i1}^k, ..., x_{in}^k)$$  \hspace{1cm} (5.1)

is an $n$-dimensional vector such that

$$\sum_{j=1}^{n} x_{ij}^k = 1 \quad \text{for all } i = 1,..,n$$  \hspace{1cm} (5.2)

$$x_{ij}^k \geq 0 \quad \text{for all } i = 1,..,n; j = 1,..,n; k = 1,..,m$$  \hspace{1cm} (5.3)

and

$$\sum_{k=1}^{m} \sum_{j=1}^{n} \phi_{ij}^k x_{ij}^k \leq p_i \quad \text{for all } i = 1,..n$$  \hspace{1cm} (5.4)

An $nn$-dimensional vector $x_i^k$ is called the strategy profile of user $k$ where

$$x_i^k = (x_{i1}^k, ..., x_{in}^k)$$  \hspace{1cm} (5.5)

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

$$x = (x_1^1, ..., x_n^1; x_1^2, ..., x_n^2; ....; x_1^m, ..., x_n^m)$$

or

$$x = (x^1, ..., x^m)$$  \hspace{1cm} (5.6)
If each node is modelled as an M/M/1 queuing system \([41]\), then the expected node delay at node \(i\) is as follows

\[
F_i(x) = \frac{1}{\mu_i - \beta_i}
\]  

(5.7)

where \(\beta_i\) is the load on node \(i\) and given as

\[
\beta_i = \sum_{k=1}^{m} \sum_{j=1}^{n} \phi_{jk} x_{jk}^k
\]

(5.8)

Clearly, \(F_i(x)\) is a strictly increasing, convex, and continuously differentiable function of \(x^j\) (\(j = 1, \ldots, m\)).

We assume as in \([81]\), 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 \(\lambda\), where

\[
\lambda = \sum_{j=1}^{m} \lambda^j
\]

(5.9)

and \(\lambda^j\) is the traffic through the network due to user \(j\) jobs given as follows
Chapter 5: Competitive Equilibrium Approach for Load Balancing a Computational Grid with Communication Delays

\[
\lambda^j = \frac{1}{2} \sum_{i=1}^{n} \left| \phi^i_j - \beta^i_j \right| 
\]

(5.10)

where \( \beta^i_j \) is the contribution on the load of node \( i \) by user \( j \) jobs given as

\[
\beta^i_j = \sum_{k=1}^{n} \phi^i_k x^j_{ki} 
\]

(5.11)

If the communication network is modelled as an M/M/1 queuing system [41], the expected communication delay of any job is given as

\[
G(\lambda) = \frac{t}{1 - t \sum_{k=1}^{m} \lambda^k} 
\]

(5.12)

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 \( \lambda \).

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

\[
T^j(\mathbf{x}) = \frac{1}{\phi^j} \sum_{i=1}^{n} \beta^i_j F_i(\mathbf{x}) + \frac{\lambda^j}{\phi^j} G(\lambda) 
\]

(5.13)

The mean response time of all jobs is given by

\[
T(\mathbf{x}) = \frac{1}{\phi} \sum_{j=1}^{m} \phi^j T^j(\mathbf{x}) 
\]

(5.14)
The best response time for user $j$ job is a solution to the following optimization problem

$$\min_{\mathbf{x}} T^j(\mathbf{x})$$  \hspace{1cm} (5.15)

subject to the constraints (5.2) to (5.4).

### 5.2. 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 $w_i \geq 0$ and use it to purchase computing resources. However, $w_i$ does not represent real money, but artificial and can be interpreted as “importance weight”. If $w_i = 1$ 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 $p_i$, where $p_i$ like $w_i$ 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 \( p = (p_1, \ldots, p_n) \) and load fractions \( x = (x_1, \ldots, x_m) \) such that \( X \) is a maximizer of

\[
\max_{X_j} u_j(x) \quad \text{for all } j = 1, \ldots, m \tag{5.16}
\]

subject to the constraints (5.2) to (5.4) and market clearing condition given by

\[
\sum_{i=1}^{n} p_i * \beta_{ij} = w_j \quad \text{for all } j = 1, \ldots, m \tag{5.17}
\]

where \( u_j(x) \) is strictly continuous, concave, and continuously differentiable function of \( x_j \) \((j = 1, \ldots, m)\). Also \( x_j \subseteq \mathbb{R}_+^n \) and a closed convex set, bounded from below. According to the lemma of abstract economy developed by Debreu [25], 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 tâtonnement trail and error introduced by Walras [54]. 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.

### 5.2.1. Algorithm (CES)

**Input**
- Node Processing Rates: $\mu_1, \ldots, \mu_n$
- Job Arrival Rates: $\phi_{ij}$ (for all $j = 1, \ldots, m; i = 1, \ldots, n$)

**Output**
- Load Fractions $x^1, \ldots, x^m$

1. **Initialization**
   1.1. $w_j \rightarrow 1$ for all $j = 1, \ldots, m$
   1.2. $p_i \rightarrow \frac{1}{n}$ for all $i = 1, \ldots, n$

2. **Loop**
   2.1. At prices $p_1, \ldots, p_n$ compute $x^1, \ldots, x^m$ such that each user maximizes her utility function (5.16) subject to the constraints (5.2) to (5.4).
   2.2. Obtain market clearing error, $\alpha$ given as follows
\[ \alpha = \sqrt{\frac{\sum_{j=1}^{m} \xi_j^2}{\sum_{j=1}^{n} \alpha \xi_j}} \]  

(5.18)

where \( \xi_j \) is given by

\[ \xi_j = w_j - \sum_{i=1}^{n} p_i \beta_i \]

(5.19)

2.3. Adjust the prices \( p_1, \ldots, p_n \) in proportion to aggregate demands

Until \( \alpha \leq \text{error tolerance} \)

5.3. 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

\[ \text{FI} = \frac{\left( \sum_{i=1}^{m} T_i(x) \right)^2}{n \sum_{i=1}^{m} (T_i(x))^2} \]

(5.20)

If \( \text{FI} = 1 \), the system is 100% fair to all users. \( \text{FI} \) decreases when, differences on \( T_i(x) \) increases and the load balancing scheme favours only few users.

The other two schemes are described below-

1. Global Optimal Solution (GOS) – In this the expected mean response time of all user jobs is minimized. The loads
Chapter 5: Competitive Equilibrium Approach for Load Balancing a Computational Grid with Communication Delays

\[ \beta_i^j \text{ (for all } j = 1,\ldots,m; i = 1,\ldots,n) \text{ are obtained by solving the following optimization problem} \]

\[
\min_{\mathbf{x}} T(\mathbf{x}) \tag{5.21}
\]

subject to the constraints (5.2) to (5.4).

2. Nash Equilibrium Solution (NES) – The loads \( \beta_i^j \text{ (for all } j = 1,\ldots,m; i = 1,\ldots,n) \) for Nash equilibrium solution are obtained by solving the optimization problem given by (5.15) (for all \( j = 1,\ldots,m \)), subject to the constraints (5.2) to (5.4).

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 (5.15) 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 \( \mathbf{x}^i^\star \) when the other user’s strategies are fixed.

5.3.1. 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.
Chapter 5: Competitive Equilibrium Approach for Load Balancing a Computational Grid with Communication Delays

### 5.3.1.1. Effect of System Utilization

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

\[
\rho = \frac{\varphi}{\sum_{i=1}^{n} \mu_i}
\]  

(5.22)

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 5.1. For a given system utilization, total job arrival rate \( \varphi \) is obtained from (5.22) above. From the \( \varphi \) obtained, the job arrival rate of user \( j \) job \( (\varphi_j^j) \) is determined from the total job arrival rate \( \varphi \) as \( \varphi_j^j = \varphi \times q_j^j \), where \( q_j^j \), the job arrival fraction of user \( j \) is given in Table 5.2.

#### Table 5.1 System Configuration

<table>
<thead>
<tr>
<th>Number of Computers</th>
<th>5</th>
<th>5</th>
<th>4</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Rate (jobs/sec)</td>
<td>10</td>
<td>20</td>
<td>50</td>
<td>100</td>
</tr>
</tbody>
</table>

The job arrival rate of user \( j \) jobs to each computer \( i \) \((i = 1,..,n)\) is determined as \( \varphi_i^j = \varphi^j \times \frac{\mu_i}{\sum_{i=1}^{n} \mu_i} \). The mean communication time \( t \) is taken to be 0.01sec.

#### Table 5.2 Job arrival fractions \( q_j^j \) of each user

<table>
<thead>
<tr>
<th>User</th>
<th>1</th>
<th>2</th>
<th>3-6</th>
<th>7-9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_j )</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.01</td>
<td>0.07</td>
</tr>
</tbody>
</table>
Figures 5.1 and 5.2 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 5.1 System Utilization Vs Mean Response Time of all User Jobs](image)

Figures 5.3, 5.4, and 5.5 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.
5.3.1.2. 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 5.3 and presented in Figures 5.6 and 5.7 respectively for system utilization of 50%.
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.
Table 5.3 System Parameters

<table>
<thead>
<tr>
<th>Speed Skewness</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_i$ of $c_1 \cdot c_2$</td>
<td>20</td>
<td>40</td>
<td>60</td>
<td>80</td>
<td>100</td>
<td>120</td>
</tr>
<tr>
<td>$\mu_i$ of $c_3 \cdot c_{16}$</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure 5.6 Heterogeneity Vs Mean Response Time of all User Jobs

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.

Figure 5.7 Heterogeneity Vs Fairness Index of the System
5.4. 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.