ANOMALOUS RELATIONS AMONG VARIOUS PERFORMANCE OBJECTIVES IN DISTRIBUTED COMPUTER SYSTEMS[1]

Hisao Kameda, Takayuki Kozawa, and Jie Li

Institute of Information Sciences and Electronics, University of Tsukuba
Tsukuba Science City, Ibaraki 305, Japan

Abstract

Distributed computer systems consist of a set of heterogeneous host computers (i.e., nodes) connected by a communication network. A job that arrives at a node may either be processed locally or transferred to another node for remote processing, which we call load balancing. One possible performance objective of load balancing in distributed computer systems is to minimize the overall mean response time. We can characterize analytically the static load balancing policy whereby the mean overall response time is minimized, which we call the overall optimal policy. This policy, however, lacks fairness in the sense that, for example, two jobs arriving at the same node but being forwarded to different nodes may not have the same expected response time. To satisfy fairness among jobs we can consider an individually optimal load balancing policy whereby jobs arriving at the same node have the same (minimum) expected response time regardless of the nodes which process them. Furthermore, we can think of a node optimal load balancing policy whereby the mean response time of jobs arriving at each node is minimum given the decision by the other nodes of which jobs arriving at those nodes are forwarded. We report the existence of some seemingly anomalous phenomena in the mutual relation among the above policies.

1 Introduction

It is very important to design and implement computer systems that have good performance. We may have, however, several performance measures in evaluating the performance of computer systems. To evaluate and optimize the performance of each computer system, we are forced to use one out of these performance measures. It is clear that the following situation could happen: Suppose that there are two computer systems A and B and that we think of two performance measures M and N. It could happen that A is better than B in performance measure M whereas A is worse than B in performance measure N. In this paper we investigate this kind of issues further. Although this kind of discussion applies to systems in general, we use distributed computer systems as the platform of our discussion.

We consider a distributed computer system that consists of a number of host computers (nodes) connected by a single-channel communication network (e.g., an Ethernet). In this paper we use the two terms, node and host, interchangeably as having the same meaning. In this system, jobs that arrive at a node can be processed locally or be forwarded to another node for remote processing. In this way, loads on nodes are balanced so as to improve the performance of the entire system. We call such balancing the load over the system the *load balancing*. Load balancing policies can be either static or adaptive (dynamic). Adaptive load balancing policies are generally based on current information about the state of the system. Static load balancing policies, on the other hand, are based on the time-average behavior of the system. Adaptive load balancing is generally more effective but more expensive than static load balancing. Since adaptive load balancing is mathematically intractable in general, we use in this paper static load balancing as a base of our discussion. The reader is referred to [1] as one example of discussions on static vs. adaptive load balancing issues.

It seems that the most common performance measure is the *overall mean response time*, which is defined to be the expected value of the time length that starts when a job arrives at the system (i.e., an arbitrary node) and ends when the job leaves the system after the processing of the job is completed. We call an optimal load balancing policy whereby the overall mean response time is minimized the *overall optimal policy.* We can formulate a nonlinear optimization problem in which the overall mean response time is minimized. The solution of the problem characterizes the decision by the overall optimal policy. We call the solution the *overall optimum.*

[1]In: *Proceedings of the 1st IEEE World Congress on Systems Simulation*, Singapore, 1997, pp.459–465

Based on the solution we can obtain an optimal load balancing algorithm which gives the decision by the overall optimal policy. In the decision the following situation can arise: The expected response time for a job that arrives at the node and is processed locally is different from the expected response time for a job that arrives at the same node but is forwarded to another node for remote processing. The users of jobs which have longer mean response times would feel this situation unfair.

On the other hand, we can think of another static load balancing policy whereby the expected response time for all jobs that arrive at the same node are identical regardless of which node processes them. Naturally there may be some nodes to which no jobs are forwarded from the node. If a job were forwarded to such nodes, the expected response time for a job would be longer than the identical one. We call such a load balancing policy an *individually optimal policy.* By the *individual optimization* problem we mean the problem of obtaining the scheduling decision that achieves the objective of the individually optimal policy. We call the solution of the individual optimization problem the *individually optimal solution* or the inter-individual *equilibrium.* We also call the solution the *individual optimum.* In the equilibrium, no user has any incentive to change the processing node of his job. The overall mean response time, however, may not be minimized by the policy. The existence of the equilibrium is proved by Kameda and Hazeyama [2] and its uniqueness is discussed by Kameda and Zhang [3].

Furthermore, we can classify arriving jobs into a finite number of classes. For example, jobs arriving at the same node can be classified into the same class. We can think that each such class corresponds to a group of users. A group of users can think of optimizing a performance measure pertaining to its own class. A group of users corresponds to a user defined by Orda et al. [4]. Each group of users decides load balancing only on its own jobs, i.e., each group of users determines whether a job of its class should be processed locally or scheduled to a different node for remote processing, given the scheduling decision on jobs of other classes. Each group of users aims at minimizing the expected response time for a job of its own class. We call the above minimization problem a *class optimization* problem. We call its solution a *class optimal solution* or an *inter-class equilibrium.* Such an equilibrium is a type of Nash equilibrium of a noncooperative game. In the inter-class equilibrium, no group of users would find it beneficial to change its job scheduling decision. We call the solution the *class optimum.* Naturally, the overall, individually, and class optimal solutions are not mutually identical.

In this paper, we consider a class optimal policy where each class has one-to-one correspondence to each node. Then we think of a *node optimal policy*, the *node optimal solution*, and the *node optimum.* We examine, by using numerical algorithms, the overall, individually, and node optimal load balancing policies. We found that according as the values of system parameters change, the decisions by these policies show sometimes awkward or seemingly anomalous behaviors. These behaviors seem to betray our intuition or expectation that if individuals or groups of users seek their own goals, the entire system improves also. Section 2 describes the model and problem formulation. Section 3 provides the numerical experiments. Section 4 presents the results and discussion. Section 5 concludes the paper.

2 The Model and Problem Formulation

The model used is that of a distributed computer system as given by Tantawi and Towsley [5], and Kim and Kameda [6][7]. The system consist of n nodes, which represent host computers, connected by a single channel communication network. Nodes may be heterogeneous; that is, they may have different configurations, number of resources, and speed characteristics. However, they have the same processing capability, that is, a job may be processed from start to finish at any node in the system.

Jobs are classified into M classes. Class-k jobs arrive at node i according to a time-invariant Poisson process with rate $\phi_i^{(k)}$. A job arriving at node i (referred to as the *origin node*) may be either processed at node i or transferred through the communication channel to another node j (processing node). After the job is processed at node j, a response (answer) is sent back to the origin node.

As the node models, we use central-server type models. As the model of the communication channel, we use an M/G/1 type model.

The notation is given as follows:

- $\phi_i^{(k)}$ External class k job arrival rate to node i
- $x_{ij}^{(k)}$ Class k job forwarding rate from node i to node j, i.e., the rate of how many class k jobs arriving at node i are forwarded to node j during unit time interval.
- $\beta_i^{(k)}$ Class k job processing rate (load) of node i $(\beta_i^{(k)} = \sum_{i=1}^{n} x_{ij}^{(k)})$
- β_i Total job processing rate (load) at node i ($\beta_i = \sum_{k=1}^{m} \beta_i^{(k)}$), i.e., the rate how many class k jobs are processed by node i during unit time interval.
- $\lambda^{(k)}$ Class k job traffic through network
- λ Total job traffic through network
- ϕ_i Total job arrival rate to node i ($\phi_i = \sum_{k=1}^{m} \phi_i^{(k)}$)
- $\phi^{(k)}$ Total class k job arrival rate ($\phi^{(k)} = \sum_{i=1}^{n} \phi_i^{(k)}$)
- Φ Total external job arrival rate ($\Phi = \sum_{k=1}^{m} \phi^{(k)}$)

- $\boldsymbol{\phi}_i$ $[\phi_i^{(1)}, \phi_i^{(2)}, \ldots, \phi_i^{(m)}]$
- $\boldsymbol{\phi}$ $[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n]$
- $\boldsymbol{\phi}^{(k)}$ $[\phi_1^{(k)}, \phi_2^{(k)}, \ldots, \phi_n^{(k)}]$
- $\boldsymbol{x}^{(k)}$ $[\boldsymbol{x}_1^{(k)}, \boldsymbol{x}_2^{(k)}, \ldots, \boldsymbol{x}_n^{(k)}]$
- $\boldsymbol{x}$ $[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}]$
- $\boldsymbol{x}_i^{(k)}$ $[x_{i1}^{(k)}, x_{i2}^{(k)}, \ldots, x_{in}^{(k)}]$
- $\boldsymbol{\beta}_i$ $[\beta_i^{(1)}, \beta_i^{(2)}, \ldots, \beta_i^{(m)}]$
- $\boldsymbol{\beta}$ $[\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_n]$
- $\boldsymbol{\beta}^k$ $[\beta_1^{(k)}, \beta_2^{(k)}, \ldots, \beta_n^{(k)}]$
- $\boldsymbol{\lambda}$ $[\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(m)}]$
- $F_i^{(k)}(\boldsymbol{\beta}_i)$ Expected node delay of class k job processed at node i (We assume that it is differentiable, increasing, and convex and that it is strictly convex as a function of a single variable $\beta_i^{(k)}$ for each k).
- $G^{(k)}(\boldsymbol{\lambda})$ Expected communication delay of class k job (We assume that it is source-destination-independent, differentiable, nondecreasing, and convex).

2.1 The objective of each policy

(1) The objective of the overall optimal policy
The objective of this policy is to implement a solution $\boldsymbol{x}$ of the following problem of minimizing the overall mean response time:
minimize

$$T(\boldsymbol{x}) = \frac{1}{\Phi}\sum_{k=1}^{m}\left(\sum_{i=1}^{n}\beta_i^{(k)}F_i^{(k)}(\boldsymbol{\beta}_i) + \lambda^{(k)}G^{(k)}(\boldsymbol{\lambda})\right),$$

subject to

$$\sum_{i=1}^{n}\beta_i^{(k)} = \phi^{(k)}, \quad k = 1, 2, \ldots, m,$$

$$\beta_i^{(k)} \geq 0, \quad i = 1, 2, \ldots, n, k = 1, 2, \ldots, m,$$

with respect to $\boldsymbol{x}$.

(2) The objective of the class optimal policy
The objective of this policy is to implement a solution $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(m)}]$ such that each $\boldsymbol{x}^{(k)}$ is a solution of the following problem of minimizing the mean response time of class k jobs given other $\boldsymbol{x}^{(j)}$, $j \neq k$:
minimize

$$T^{(k)}(\boldsymbol{x}) = \frac{1}{\phi^{(k)}}\left(\sum_{i=1}^{n}\beta_i^{(k)}F_i^{(k)}(\boldsymbol{\beta}_i) + \lambda^{(k)}G^{(k)}(\boldsymbol{\lambda})\right),$$

subject to

$$\sum_{i=1}^{n}\beta_i^{(k)} = \phi^{(k)},$$

$$\beta_i^{(k)} \geq 0, \quad i = 1, 2, \ldots, n,$$

with respect to $\boldsymbol{x}^{(k)}$ given other $\boldsymbol{x}^{(j)}$, $j \neq k$.

The objective of the node optimal policy can be obtained by identifying class index k with node index i.

(3) The objective of the individually optimal policy
The objective of this policy is to implement such a solution $\boldsymbol{x}$ that for each class k jobs that arrive at node i, if there exist jobs that are processed at node j, the following relation holds:

$$T_{ij}^{(k)}(\boldsymbol{x}) = \min_{l} T_{il}^{(k)}(\boldsymbol{x}).$$

2.2 The solution of each policy

(1) The solution of the overall optimal policy
Define $f_i^{(k)}(\boldsymbol{\beta}_i)$, $g^{(k)}(\boldsymbol{\lambda})$ and $f_i^{(k)-1}(\boldsymbol{\beta}_i\,|_{\beta_i^{(k)}=x})$ as follows.

$$f_i^{(k)}(\boldsymbol{\beta}_i) = \frac{\partial}{\partial\beta_i^{(k)}}\sum_{l=1}^{m}\beta_i^{(l)}F_i^{(l)}(\boldsymbol{\beta}_i).$$

$$g^{(k)}(\boldsymbol{\lambda}) = \frac{\partial}{\partial\lambda^{(k)}}\sum_{l=1}^{m}\lambda^{(l)}G^{(l)}(\boldsymbol{\lambda}).$$

$$f_i^{(k)-1}(\boldsymbol{\beta}_i\,|_{\beta_i^{(k)}=x}) = \begin{cases} a & f_i^{(k)}(\boldsymbol{\beta}_i\,|_{\beta_i^{(k)}=\alpha}) = x \\ 0 & f_i^{(k)}(\boldsymbol{\beta}_i\,|_{\beta_i^{(k)}=0}) \geq x \end{cases}$$

where $\boldsymbol{\beta}_i\,|_{\beta_i^{(k)}=x}$ denotes the vector whose elements are the sames as those of $\boldsymbol{\beta}_i$ except that the element $\beta_i^{(k)}$ is replaced by x.

The decision of $\boldsymbol{\beta}$ by the overall optimal policy is given by the following relations [6].

$f_i^{(k)}(\boldsymbol{\beta}_i) \geq \alpha^{(k)} + g^{(k)}(\boldsymbol{\lambda})$ and $\beta_i^{(k)} = 0$ $(i \in R_d^{(k)})$
or
$f_i^{(k)}(\boldsymbol{\beta}_i) = \alpha^{(k)} + g^{(k)}(\boldsymbol{\lambda})$ and $0 < \beta_i^{(k)} < \phi_i^{(k)}$ $(i \in R_a^{(k)})$
or
$\alpha^{(k)} \geq f_i^{(k)}(\boldsymbol{\beta}_i) \geq \alpha^{(k)} + g^{(k)}(\boldsymbol{\lambda})$ and $\beta_i^{(k)} = \phi_i^{(k)}$ $(i \in N^{(k)})$
or
$\alpha^{(k)} = f_i^{(k)}(\boldsymbol{\beta}_i)$ and $\beta_i^{(k)} \leq \phi_i^{(k)}$ $(i \in S^{(k)})$
subject to

$$\sum_{i\in S^{(k)}} f_i^{(k)}(\boldsymbol{\beta}_i\,|_{\beta_i^{(k)}=\alpha^{(k)}}) + \sum_{i\in N^{(k)}}\phi_i^{(k)} + \sum_{i\in R_a^{(k)}} f_i^{(k)-1}(\boldsymbol{\beta}_i\,|_{\beta_i^{(k)}=\alpha^{(k)}+g^{(k)}(\lambda)}) = \phi^{(k)},$$

where $\alpha^{(k)}$ is the *Lagrange multiplier.*

(2) The solution of the class optimal policy
Define $\tilde{f}_i^{(k)}$,$\tilde{g}^{(k)}$ as follows.

$$\tilde{f}_i^{(k)}(\boldsymbol{\beta}_i) = \frac{\partial}{\partial\beta_i^{(k)}}\beta_i^{(k)}F_i^{(k)}(\boldsymbol{\beta}_i),$$

$$\tilde{g}^{(k)}(\boldsymbol{\lambda}) = \frac{\partial}{\partial \lambda^{(k)}} \lambda^{(k)} G^{(k)}(\boldsymbol{\lambda}).$$

Define $\tilde{f}_i^{(k)-1}$ similarly as $f_i^{(k)-1}$. Then the decision of $\boldsymbol{\beta}$ by the class optimal policy is given by the following relations, similarly as the above [2].

$\tilde{f}_i^{(k)}(\boldsymbol{\beta}_i) \geq \tilde{\alpha}^{(k)} + \tilde{g}^{(k)}(\boldsymbol{\lambda})$ and $\beta_i^{(k)} = 0$ $(i \in R_d^{(k)})$
or
$\tilde{f}_i^{(k)}(\boldsymbol{\beta}_i) = \tilde{\alpha}^{(k)} + \tilde{g}^{(k)}(\boldsymbol{\lambda})$ and $0 < \beta_i^{(k)} < \phi_i^{(k)}$ $(i \in R_a^{(k)})$
or
$\tilde{\alpha}^{(k)} \geq \tilde{f}_i^{(k)}(\boldsymbol{\beta}_i) \geq \tilde{\alpha}^{(k)} + \tilde{g}^{(k)}(\boldsymbol{\lambda})$ and $\beta_i^{(k)} = \phi_i^{(k)}$ $(i \in N^{(k)})$
or
$\tilde{\alpha}^{(k)} = \tilde{f}_i^{(k)}(\boldsymbol{\beta}_i)$ and $\beta_i^{(k)} \leq \phi_i^{(k)}$ $(i \in S^{(k)})$
subject to

$$\sum_{i \in S^{(k)}} F_i^{(k)}(\boldsymbol{\beta}_i \mid_{\beta_i^{(k)} = \tilde{\alpha}^{(k)}}) + \sum_{i \in N^{(k)}} \phi_i^{(k)} + \sum_{i \in R_a^{(k)}} \tilde{f}_i^{(k)-1}(\boldsymbol{\beta}_i \mid_{\beta_i^{(k)} = \tilde{\alpha}^{(k)} + \tilde{g}^{(k)}(\lambda)}) = \phi^{(k)},$$

where $\tilde{\alpha}^{(k)}$ is the *Lagrange multiplier*.

The solution of the node optimal policy can be obtained by identifying class index k with node index i.

(3) The solution of the individually optimal policy

Define $F_i^{(k)-1}$ similarly as $f_i^{(k)-1}$. Then, the decision of $\boldsymbol{\beta}$ by the individually optimal policy is given by the following relations, similarly as the above [2].

$F_i^{(k)}(\boldsymbol{\beta}_i) \geq K^{(k)} + G^{(k)}(\boldsymbol{\lambda})$ and $\beta_i^{(k)} = 0$ $(i \in R_d^{(k)})$
or
$F_i^{(k)}(\boldsymbol{\beta}_i) = K^{(k)} + G^{(k)}(\boldsymbol{\lambda})$ and $0 < \beta_i^{(k)} < \phi_i^{(k)}$ $(i \in R_a^{(k)})$
or
$K^{(k)} \geq F_i^{(k)}(\boldsymbol{\beta}_i) \geq K^{(k)} + G^{(k)}(\boldsymbol{\lambda})$ and $\beta_i^{(k)} = \phi_i^{(k)}$ $(i \in N^{(k)})$
or
$K^{(k)} = F_i^{(k)}(\boldsymbol{\beta}_i)$ and $\beta_i^{(k)} \leq \phi_i^{(k)}$ $(i \in S^{(k)})$
subject to

$$\sum_{i \in S^{(k)}} F_i^{(k)}(\boldsymbol{\beta}_i \mid_{\beta_i^{(k)} = K^{(k)}}) + \sum_{i \in N^{(k)}} \phi_i^{(k)} + \sum_{i \in R_a^{(k)}} F_i^{(k)-1}(\boldsymbol{\beta}_i \mid_{\beta_i^{(k)} = K^{(k)} + G^{(k)}(\lambda)}) = \phi^{(k)},$$

where $K^{(k)} = \min_i F_i^{(k)}(\boldsymbol{\beta}_i)$. Therefore the decisions by these three optimal policies can be obtained by the Kim and Kameda algorithm [6] or the slightly modified version of it. $\boldsymbol{x}$ can be obtained form $\boldsymbol{\beta}$.

In case $\beta_i^{(k)} = 0 (i \in R_d^{(k)})$, $x_{ii}^{(k)} = 0$ and $x_{ij}^{(k)} \geq 0$ $(j \in S^{(k)})$, $\sum_{j \in S^{(k)}} x_{ij} = \phi_i^{(k)}$. In case $0 < \beta_i^{(k)} < \phi_i^{(k)} (i \in R_a^{(k)})$, $x_{ii}^{(k)} = \beta_i^{(k)}$ and $x_{ij}^{(k)} \geq 0$ $(j \in S^{(k)})$, $\sum_{j \in S^{(k)}} x_{ij} = \phi_i^{(k)} - \beta_i^{(k)}$. In case $\beta_i^{(k)} = \phi_i^{(k)} (i \in N^{(k)})$, $x_{ii}^{(k)} = \phi_i^{(k)}$ and $x_{ij}^{(k)} = 0$, for all $j \neq i$. In case $\beta_i^{(k)} \geq \phi_i^{(k)} (i \in S^{(k)})$, $x_{ii}^{(k)} = \phi_i^{(k)}$ and $x_{ij}^{(k)} = 0$, for all $j \neq i$.

3 Numerical Experiments

In this section, we give the details of the system model that we use in the numerical experiments. The system consists of nodes and a communication channel. We also give the values of the system parameters examined in the numerical experiments.

3.1 The node model

We use a central-server type model. Server 0 is a CPU that processes jobs according to the processor sharing discipline. Servers 1,2,...,d are I/O devices that process jobs according to the FCFS discipline. Let $p_{i,0}^{(k)}$ and $p_{i,j}^{(k)}$, j=1,2,...,d denote the transition probabilities that, after departing from the CPU, a class k job leaves the node i or requests and I/O service at device j, j=1,2,...,d, respectively.

Then the expected class k node delay is given by

$$F_i^{(k)}(\boldsymbol{\beta}_i) = \sum_{j=1}^{d} \frac{q_{i,j}^{(k)} \frac{1}{\mu_{i,j}^{(k)}}}{1 - (q_{i,j}^{(1)} \frac{\beta_i^{(1)}}{\mu_{i,j}^{(1)}} + \cdots + q_{i,j}^{(m)} \frac{\beta_i^{(m)}}{\mu_{i,j}^{(m)}})},$$

where $q_{i,0}^{(k)} = \dfrac{1}{p_{i,0}^{(k)}}$, and $q_{i,j}^{(k)} = \dfrac{p_{i,j}^{(k)}}{p_{i,0}^{(k)}}$, and where $\mu_{i,j}^{(k)}$ denotes the class k processing rate for server j at node i. $\mu_{i,j}^{(k)}$ is the same for j=1,2,...,d and k=1,2,...,m.

3.2 The communication channel model

We use a processor sharing M/G/1 model. Expected class k communication delay is given by

$$G^{(k)}(\boldsymbol{\lambda}) = \frac{t^{(k)}}{1 - (t^{(1)}\lambda^{(1)} + \cdots + t^{(m)}\lambda^{(m)})},$$

where $t^{(k)}$ denotes the *mean communication time* (excluding the queueing time) for class k jobs.

3.3 The parameters of the experiment

We examined 4 node model (n=4). We assume that each class corresponds to a node. That is, $\phi_i^{(k)}$=0, $i \neq k$. We denote $\phi_i^{(k)}$ by ϕ_i and all other system parameters are independent of the class. That is, for example, $\mu_{i,j}^{(k)} = \mu_{i,j}^{(k')} = \mu_{i,j}$, $q_{i,j}^{(k)} = q_{i,j}^{(k')} = q_{i,j}$, $t^{(k)} = t^{(k')} = t$. Therefore $F_i^{(k)}(\boldsymbol{\beta}_i) = F_i^{(k')}(\boldsymbol{\beta}_i) = F_i(\boldsymbol{\beta}_i)$, $G^{(k)}(\boldsymbol{\lambda}) = G^{(k')}(\boldsymbol{\lambda}) = G(\boldsymbol{\lambda})$. Thus we say the node optimal policy instead of the class optimal policy.

The system parameter is given as follows: (See Fig.1) (The unit of measure of u_{ij} is jobs/sec.)

$d_1 = 3$

$\mu_{1,0} = 1500.0$	$\mu_{1,1} = 450.0$	$\mu_{1,2} = 450.0$	$\mu_{1,3} = 450.0$
$q_{1,0} = 10.0$	$q_{1,1} = 3.0$	$q_{1,2} = 3.0$	$q_{1,3} = 3.0$

$d_2 = 2$

$\mu_{2,0} = 100.0$	$\mu_{2,1} = 20.0$	$\mu_{2,2} = 20.0$	
$q_{2,0} = 10.0$	$q_{2,1} = 4.5$	$q_{2,2} = 4.5$	

$d_3 = 3$

$\mu_{3,0} = 100.0$	$\mu_{3,1} = 30.0$	$\mu_{3,2} = 30.0$	$\mu_{3,3} = 30.0$
$q_{3,0} = 10.0$	$q_{3,1} = 3.0$	$q_{3,2} = 3.0$	$q_{3,3} = 3.0$

$d_4 = 2$

$\mu_{4,0} = 120.0$	$\mu_{4,1} = 54.0$	$\mu_{4,2} = 54.0$	
$q_{4,0} = 10.0$	$q_{4,1} = 4.5$	$q_{4,2} = 4.5$	

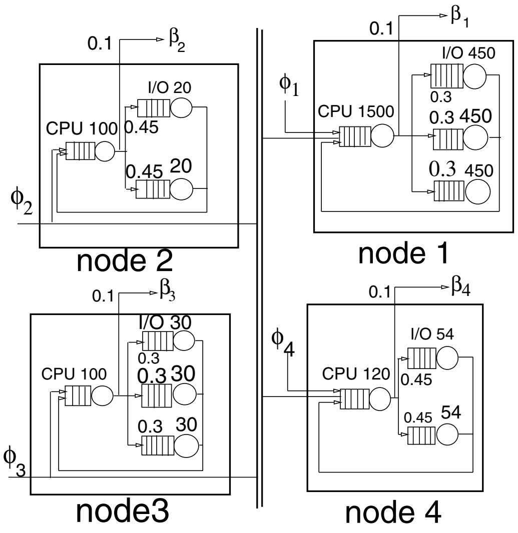

Figure 1: A distributed computer system model.

We first obtain the case (A) where
$\phi_1 = 80.0, \phi_2 = 7.0, \phi_3 = 7.0, \phi_4 = 7.5$ $(jobs/sec)$,
for $0 \leq t < 4.00$,
and the case (B) where
$\phi_1 = 120.0, \phi_2 = 7.0, \phi_3 = 7.0, \phi_4 = 7.5$ $(jobs/sec)$,
for $0 \leq t < 4.00$,

In both cases of A and B, we apply three load balancing algorithms: the overall optimal algorithm, the individually optimal algorithm, and the class(=node) optimal algorithm, and obtain the values of performance variables such as the overall mean response time, the mean response time for jobs arriving at each node, etc. These algorithms are obtained from the Kim and Kameda algorithm [6] as mentioned at the end of section 3.

4 Results and Discussion

Figures 2, 3, . . . , 9 show the results of calculation. Figures 2 and 3 show how the overall mean response time of the overall optimum, the individual optimum, and the node optimum depend on the value of mean communication time, respectively, in cases A and B. It is natural that we conjecture that the larger the mean communication time, the less the merit of forwarding jobs to other nodes for each policy.

In Figure 2 (case A), the effect of the mean communication time on the overall mean response time is similar for three optimal policies. That is, the behaviors of these three policies resemble to one another, although the values of the mean response time is generally different for each optimum. This seems in line with what we intuitively guess; the individual optimization and the node optimization will lead to an approximate overall optimization. Figures 4, 5, and 6 show the effect of the communication time on the job forwarding rates in case A, under the overall, individually, and node optimal policies. We can see that the rate of jobs forwarded decreases for each source node (nodes 1, 2, and 3) as the mean communication time increases.

Occasionally, however, we observed such counterintuitive cases as illustrated in Fig. 3 (case B). In Figure 3, the effect of the communication time on the overall mean response time seems awkward for individual and node optimum. The mean response time under the individually optimal policy is very large for very small mean communication time, and it first decreases and then increases as the mean communication time increases. The effect of the mean communication time on the mean response time under the node optimal policy appears to be more strange. As the mean communication time increases, it first increases, then decreases, and again increases. The above phenomenon tells us that these three policies have clearly different effects.

Figures 7, 8, and 9 show the effect of the communication time on the job forwarding in case B, under the overall, individually, and node optimal policies, respectively. We observed the following seemingly counterintuitive case peculiar to the node optimum. In Figure 9 (case B), under the node optimal policy, we can see that, for small t, a part of jobs arriving at nodes 2, 3 and 4 are forwarded to node 1 for remote processing, that is, x_{21}, x_{31}, and x_{41} are not zero. In return, however, a part of jobs arriving at node 1 are forwarded to nodes 2, 3 and 4 for remote processing, that is, x_{12}, x_{13}, and x_{14} are not zero. It is clear that this type of mutual forwarding never leads to the overall optimum. It is natural that different policies behave differently from one another. These types of the phenomena look, however, remarkably awkward. We should anticipate these strange behaviors in some occasions.

5 Conclusion

We have studied the mutual relationship among various performance objectives. We used the distributed computer system as the basis of our discussion. We examined three distinct optimal policies, the overall, individually, and node optimal policies. By means of numerical experiment, we have found that seemingly anomalous behaviors among these performance objectives may sometimes occur.

References

[1] H. Kameda, J. Li, C. Kim and Y. Zhang: *Optimal Load Balancing in Distributed Computer Systems*, Springer, 1996.

[2] H. Kameda and A. Hazeyama: "Individual vs. overall optimization for static load balancing in distributed computer systems", *Computer Science Report, The University of Electro-Communications*, 1988.

[3] H. Kameda and Y. Zhang: "Uniqueness of the solution for in optimal static routing in open BCMP queueing networks" *Mathematical and Computer Modelling, 22*, 10-12, pp.119-130, 1995.

[4] A. Orda, R. Rom and N. Shinkin: "Competitive routing in multiuser communication networks", *IEEE/ACM Trans. Networking, Vol.* 1, No. 5, Oct. 1993. pp.510-521.

[5] A. N. Tantawi and D. Towsley: "Optimal static load balancing in distributed computer systems", *J.ACM 32*, 2, pp.445-465 April, 1985.

[6] C. Kim and H. Kameda: "Optimal static load balancing of multi-class jobs in a distributed computer system, *Proc. 10th Intl. Conf. Distributed Comput. Syst., IEEE*, 1990, pp.562-569.

[7] C. Kim and H. Kameda: "An algorithm for optimal static load balancing in distributed computer systems, *IEEE Trans. Compt.,* 41, 3, 1990, pp.381-384.

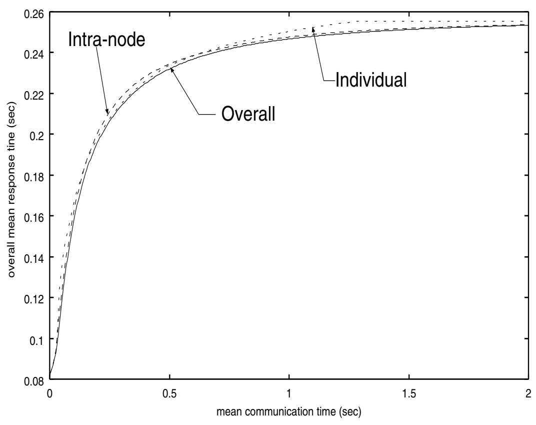

Figure 2: Overall mean response time. (case A)

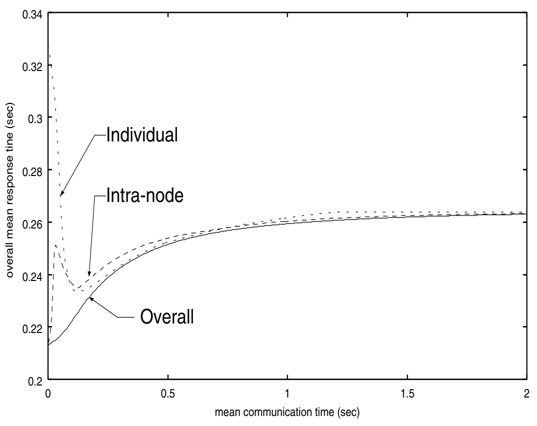

Figure 3: Overall mean response time. (case B)

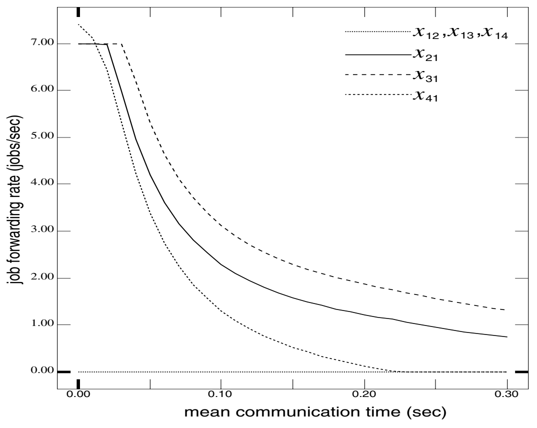

Figure 4: Job forwarding rates in the overall optimum (case A).
x_{ij} denotes the rate of jobs that arrive at node i and are forwarded to node j by the overall optimal policy.

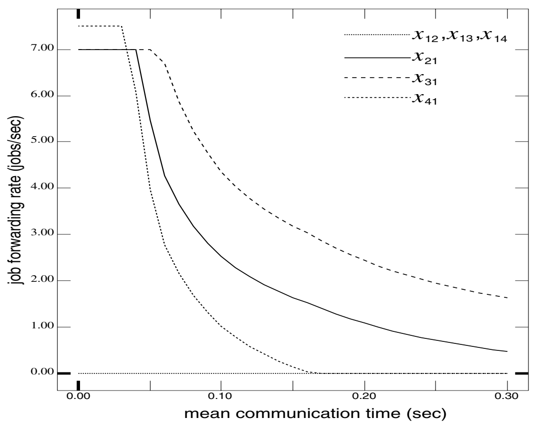

Figure 5: Job forwarding rates in the individual optimum (case A).
x_{ij} denotes the rate of jobs that arrive at node i and are forwarded to node j by the individually optimal policy.

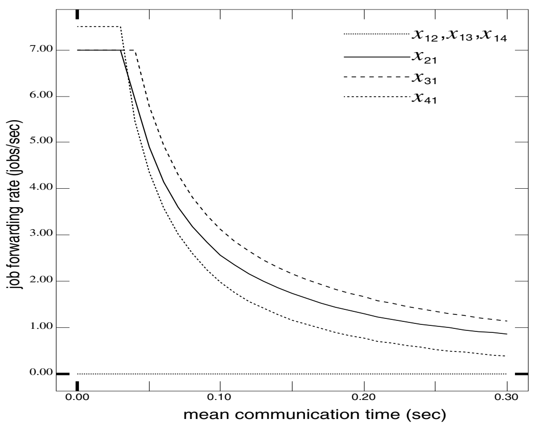

Figure 6: Job forwarding rates in the node optimum (case A).
x_{ij} denotes the rate of jobs that arrive at node i and are forwarded to node j by the node optimal policy.

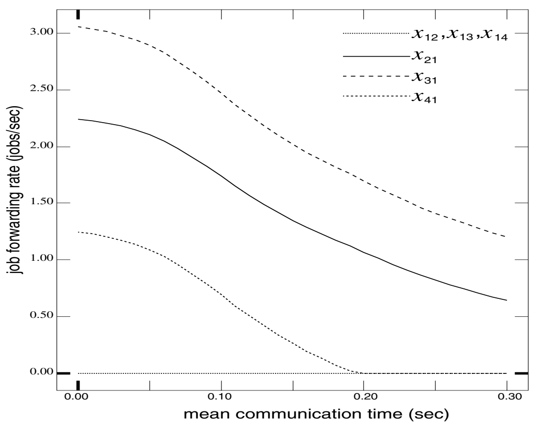

Figure 7: Job forwarding rates in the overall optimum (case B).
x_{ij} denotes the same as in Figure 4.

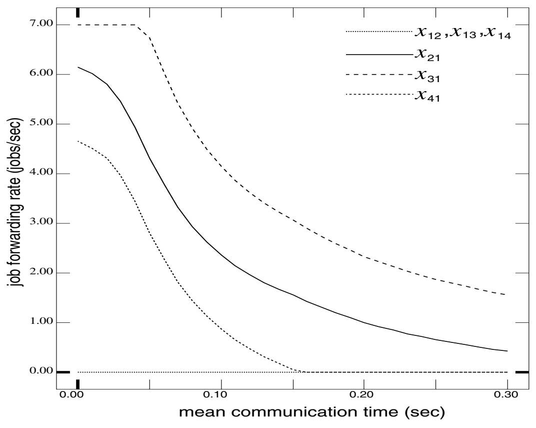

Figure 8: Job forwarding rates in the individual optimum (case B).
x_{ij} denotes the same as in Figure 5.

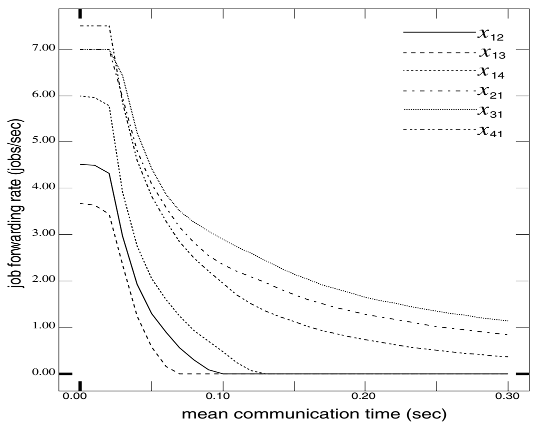

Figure 9: Job forwarding rates in the node optimum (case B).
x_{ij} denotes the same as in Figure 6.