



Available at  
**www.ElsevierMathematics.com**  
 POWERED BY SCIENCE @ DIRECT®

An International Journal  
**computers &  
 mathematics**  
 with applications

Computers and Mathematics with Applications 46 (2003) 1545–1557

www.elsevier.com/locate/camwa

# The Effect of Start-Up Delays in Scheduling Divisible Loads on Bus Networks: An Alternate Approach

S. SURESH, V. MANI\* AND S. N. OMKAR

Department of Aerospace Engineering, Indian Institute of Science  
 Bangalore 560 012, India  
 mani@aero.iisc.ernet.in

(Received December 2001; accepted January 2003)

**Abstract**—In this paper, scheduling of divisible loads in a bus network is considered. The objective is to minimize the processing time by including the overhead component due to start-up time that could degrade the performance of the system, in addition to the inherent communication and computation delays. These overheads are considered to be constant additive factors to the communication and computation components. A closed-form expression for optimal processing time is derived. Using this closed-form expression, this paper analytically proves significant results regarding the optimal sequence of load distribution and optimal number of processors. Numerical examples are presented to illustrate the analysis. © 2003 Elsevier Ltd. All rights reserved.

**Keywords**—Divisible loads, Communication delay, Processing time, Optimal sequence, Bus networks.

## NOMENCLATURE

$\alpha_i$	load fraction assigned to processor $p_i$	$z$	the inverse of the communication speed of the link
$w_i$	the inverse of the computation speed of processor $p_i$	$T_{cm}$	time taken to transmit a unit load by the communication link
$T_{cp}$	time taken to process a unit load by the standard processor	$\theta_{cm}$	a constant additive communication overhead component that includes the sum of all delays associated with the communication process
$\theta_{cp}$	a constant additive computation overhead component that includes the sum of all delays associated with the computation process		

## 1. INTRODUCTION

Scheduling problems that arise in optimally distributing the jobs among a set of available processors, with the objective of minimizing the processing time, is an important area of research in computing and communication. The prime objective in this area of research is to design efficient scheduling algorithms that minimize the total processing time [1,2]. The domain of scheduling

\*Author to whom all correspondence should be addressed.

divisible loads in multiprocessor system was started in 1988 and has stimulated considerable interest among researchers and engineers.

A divisible load can be divided into any number of fractions and can be processed independently on the processors, as there are no precedence relationships. The problem of scheduling divisible loads in a linear network incorporating the associated communication delays was first introduced in [3]. In fact, this is the first paper in this area of divisible load scheduling. In this paper, the timing diagram representation of the load distribution process, and the recursive load distribution equations were introduced. The ideas from this paper were extended to scheduling divisible loads in tree networks and bus networks in [4,5]. In these studies, the optimal load fractions are obtained by assuming all the processors involved in the computation of the load stop computing at the same time instant. In fact, this assumption has been shown to be a necessary and sufficient condition to obtain optimal processing time in linear networks [6], using the concept of processor equivalence, and an analytic proof for bus networks in [7]. However, it has been rigorously shown that this condition is true only in a restricted sense [8], in the case of a heterogeneous single-level tree networks. A closed-form expression for the processing time, for a single-level tree network is presented in [9,10] and using this closed-form expression, optimal sequence and optimal network arrangement are obtained in [9]. For the case of homogeneous linear and tree networks, closed-form expression for the processing time and an asymptotic performance analysis are carried out in [11,12]. A practical application of divisible load scheduling with reference to matrix vector products of very large size presented in [13] shows the usefulness of the analysis.

In this paper, scheduling divisible loads in bus network architecture is considered. It is possible in practical data communication and computing situations to have overheads in communication ( $\theta_{cm}$ ) and computation ( $\theta_{cp}$ ). These overheads occur in communication ( $\theta_{cm}$ ) due to protocol processing delays, unavailability of certain communication resources, and queuing delays, etc., [14,15]. Similarly, the computation overheads ( $\theta_{cp}$ ) arise due to delay in extracting the data, processor initialization, etc. [14,15]. These overheads are almost constant quantities and form as an additive component in load distribution equations [14,15]. These overheads were considered in literature by some researchers for some specific cases [15–17], such as query processing and image processing applications [15], and for different architecture [16,17]. In a recent study [18], the effect of these ‘overheads’ in the processing time is presented.

### 1.1. The Contribution of this Paper

With these overhead factors in communication and computation, we first derive a closed-form expression for the processing time. With our closed-form expression, we can obtain the processing time directly. Using this closed-form expression, we obtain the optimal number of processor and the optimal sequence of load distribution.

This paper is organized as follows. Section 2 presents the mathematical modeling and relevant definitions. In Section 3, we present the closed-form expression for the processing time and a comparison with the results obtained in earlier study [18]. Section 4 presents the optimal sequence of load distribution and Section 5 presents the Conclusion. Since this paper presents an alternative approach to the problem dealt with in an earlier study [18], for convenience, we follow the same notation used in the earlier study [18].

## 2. MATHEMATICAL MODELING AND DEFINITION

The bus network architecture considered in this paper is shown in Figure.1. This network has a dedicated bus controller unit (BCU) to distribute the entire load. The divisible load arrives at the BCU; the BCU divides the load into  $m$  fractions  $\alpha_1, \alpha_2, \dots, \alpha_m$  and distributes these load fractions to the  $m$  processor in a sequence,  $p_1, p_2, \dots, p_m$ , one after another. The processors start computing the load fractions immediately after receiving the load fractions. The objective here is to find the optimal size of these load fractions  $\alpha_1, \alpha_2, \dots, \alpha_m$ , such that the processing time is

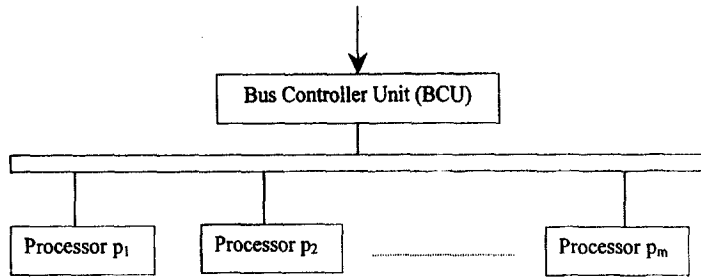


Figure 1. Distributed bus architecture with  $m$  processors and a BCU.

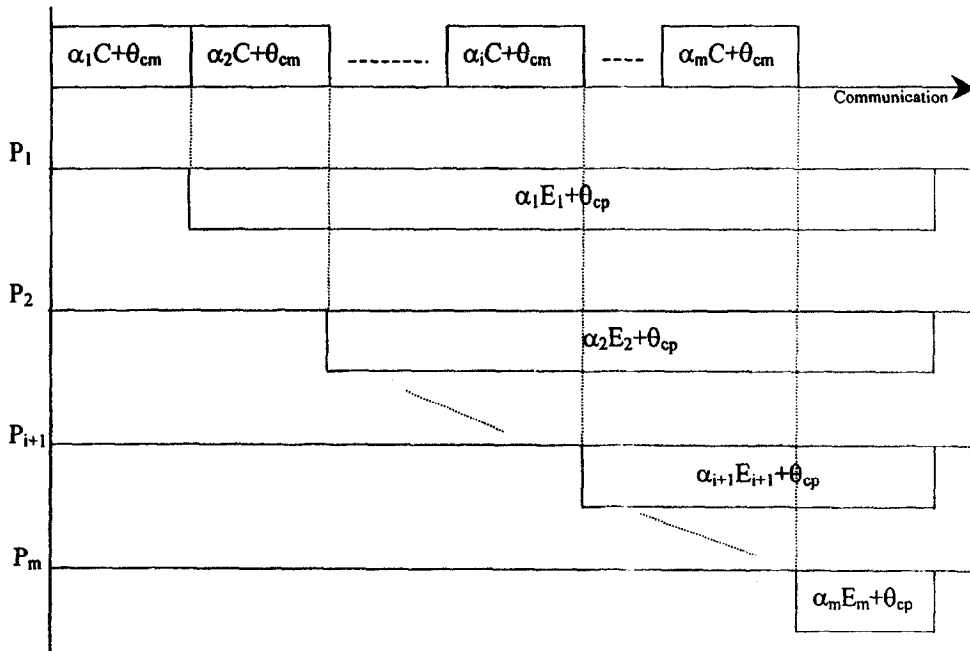


Figure 2. Timing diagram for processing the divisible load with  $m$  processors.

a minimum. As in the earlier study [18], here we denote  $w_i T_{cp}$  as  $E_i$  and  $z T_{cm}$  as  $C$ , and hence, the communication time for the load fraction  $\alpha_i$  is  $\alpha_i C + \theta_{cm}$  and the computation time for the load fraction  $\alpha_i$  is  $\alpha_i E_i + \theta_{cp}$ . The timing diagram for the load distribution process is shown in Figure 2.

1. The load distribution, denoted as  $\alpha$ , defined as an  $m$ -tuple  $(\alpha_1, \alpha_2, \dots, \alpha_m)$  such that  $0 < \alpha_i \leq 1$  and  $\sum_{i=1}^m \alpha_i = 1$ . The equation  $\sum_{i=1}^m \alpha_i = 1$  is normalization equation, and the space of all possible load distribution is denoted as  $\Gamma$ .
2. The finish time of processor  $p_i$ , denoted as  $T_i(\alpha, m)$ , is the time difference between the instant at which the  $i^{\text{th}}$  processor stops computing and the time instant at which the BCU initiates the load distribution process.
3. The processing time, denoted as  $T(\alpha, m)$ , is the time at which the entire load is processed, i.e.,  $T(\alpha, m) = \max\{T_i(\alpha, m), i = 1, 2, \dots, m\}$ , where  $T_i$  is the finish time for processor  $p_i$ .
4. The optimal processing time, denoted as  $T^*(\alpha^*, m)$ , is the minimum processing time to finish the entire load, i.e.,  $T^*(\alpha^*, m) = \min_{\alpha \in \Gamma} T(\alpha, m)$ .

In the literature [8], it has been rigorously proved that for the optimal processing time, all the processors involved in the computation of the processing load must stop computing at the same time instant. In this paper also, we use this optimality criterion.

### 3. CLOSED-FORM EXPRESSION FOR THE PROCESSING TIME

Now we shall derive a closed form for the processing time. This is derived by assuming that the sequence of load distribution is from  $p_1, p_2, \dots, p_m$  in that order. This means that the BCU unit distributes the load from processor  $p_1$  to processor  $p_m$  one after another. From the timing diagram shown in Figure 2, the recursive equations for load distribution are

$$\alpha_i E_i + \theta_{cp} = \alpha_{i+1}(E_i + C) + \theta_{cp} + \theta_{cm}, \quad i = 1, 2, \dots, m-1. \quad (1)$$

Denoting  $(E_{i+1} + C)/E_i = f_{i+1}$  and  $\theta_{cm}/E_i = \beta_i$ , for all  $i = 1, 2, \dots, m-1$ . Equation (1) can be rewritten as

$$\alpha_i = \alpha_{i+1} f_{i+1} + \beta_i, \quad i = 1, 2, \dots, m-1. \quad (2)$$

Now, we see, from the above, there are  $m-1$  linear equations with  $m$  variables, and together with the normalization equation, we have  $m$  equations. In the earlier study [18], these equations are solved as follows, to obtain the individual load fractions. Each of the  $\alpha_i$  in equation (2) is expressed in terms of  $\alpha_m$  as

$$\alpha_i = \alpha_m M_i + N_i, \quad i = 1, 2, \dots, m-1, \quad (3)$$

where

$$M_i = \prod_{j=i+1}^m f_j, \quad i = 1, 2, \dots, m-1,$$

$$N_i = \sum_{p=i}^{m-1} \beta_p \left( \prod_{j=i+1}^p f_j \right), \quad i = 1, 2, \dots, m-1,$$

and  $\alpha_m$  is obtained as

$$\alpha_m = \frac{1 - X(m)}{Y(m)}, \quad (4)$$

where

$$X(m) = \sum_{i=1}^{m-1} \sum_{p=i}^{m-1} \beta_p \left( \prod_{j=i+1}^p f_j \right)$$

and

$$Y(m) = \left( 1 + \sum_{i=1}^{m-1} \prod_{j=i+1}^m f_j \right).$$

The expression for the processing time is obtained as

$$T(\alpha, m) = \alpha_1(E_1 + C) + \theta_{cp} + \theta_{cm}. \quad (5)$$

Substituting the value of  $\alpha_1$  in the above equation,

$$T(\alpha, m) = (\alpha_m M_1 + N_1)(E_1 + C) + \theta_{cp} + \theta_{cm}, \quad (6)$$

where  $\alpha_m$ ,  $M_1$ , and  $N_1$  are defined as above.

It is shown in [18] that with the inclusion of all these overheads, for optimal processing time, it may not be necessary to use all the  $m$  processors in the system. It is shown that there exists a maximum number of processors  $m^*$  that can be utilized with the given sequence of load distribution. The necessary and sufficient condition for the existence of optimal processing time using all the  $m$  processors in a specific order is given by

$$X(m) = \sum_{i=1}^{m-1} \sum_{p=i}^{m-1} \beta_p \left( \prod_{j=i+1}^p f_j \right) < 1. \quad (7)$$

### Alternate Approach

In our (alternate) approach, the value of  $\alpha_1$  is obtained as follows. Express all the  $\alpha_i$  ( $i = 1, 2, \dots, m - 1$ ) in terms of  $\alpha_m$ . Obtain the value of  $\alpha_m$  using the normalization equation. Using this value of  $\alpha_m$ , the value of  $\alpha_1$  is obtained as

$$\alpha_1 = \frac{M_1 + Z(m)}{Y(m)}, \tag{8}$$

since  $\alpha_1$  is known, the other load fraction can be obtained as from equation (2). Hence, processing time is

$$T(\alpha, m) = (E_1 + C) \frac{M_1 + Z(m)}{Y(m)} + \theta_{cp} + \theta_{cm}, \tag{9}$$

where

$$Z(m) = \sum_{i=1}^{m-1} \beta_i \left( \prod_{k=1}^i f_k \right) \left[ 1 + \sum_{p=i+2}^m \prod_{j=p}^m f_j \right]$$

and the value of  $f_1 = 1$ . The processing time obtained in our approach is the same as the processing time obtained in the earlier approach for a given sequence of load distribution. We see in the processing time expression,  $M_1$ ,  $Z(m)$ , and  $Y(m)$  are functions of  $m$ , the number of processors. Once  $m$  is given, in our approach,  $\alpha_1$  can be directly obtained, and hence, the processing time also can be directly obtained. In the earlier study, when  $X(m) < 1$ , the processing time is obtained as follows. First, the value of  $\alpha_m$  is obtained, and then the value of  $\alpha_1$  is obtained using  $\alpha_m$ . In our approach,  $\alpha_1$  is directly obtained. While obtaining the value of  $\alpha_1$ , the necessary and sufficient condition for existence of solution  $X(m) < 1$  is not considered. It is mentioned in [18], for an  $m$ -processor system, there exists an  $m^*$  (optimal number of processors) beyond which an optimal solution ceases to exist. This is so because once this condition is not satisfied, some of the load fractions will be negative. In our closed-form expression, this violation of the necessary and sufficient condition is reflected as an increase in the value of  $\alpha_1$ . We will now show that, using the closed-form expression obtained in our approach, we can easily prove all the results obtained in the earlier study.

First, we show, in our approach, the existence of an optimal number of processors as obtained in [18]. For this purpose, we will write the value of  $\alpha_1$  obtained in our approach in the following manner:

$$\alpha_1 = \frac{M_1(m) + Z(m)}{Y(m)}. \tag{10}$$

$M_1(m)$  is the value of  $M_1$  with  $m$  processors. We know that  $M_1(m)/Y(m)$  decreases with increasing  $m$ , and  $Z(m)/Y(m)$  increases with increasing  $m$ . Hence, there is an optimal number of processors  $m^*$ , such that up to the value of  $m^*$ , the value of  $\alpha_1$  will be decreasing, and after that  $m^*$  the processing time increases in our approach. It is sufficient to prove the behavior of  $\alpha_1$  to study the behavior of the processing time. Hence,  $\alpha_1(m^*)$  has the following properties:

$$\begin{aligned} \alpha_1(m^*) &< \alpha_1(m^* - 1), \\ \alpha_1(m^*) &< \alpha_1(m^* + 1). \end{aligned} \tag{11}$$

From the earlier study, we see that the necessary and sufficient condition, for the existence of optimal processing time, with  $m^*$  processors in a specific sequence is given by  $X(m^*) < 1$ . We will now show that the  $m^*$  obtained in our approach also satisfies this condition.

**LEMMA 1.** Consider an  $m$ -processor system with a fixed sequence of load distribution as  $p_1, p_2, \dots, p_m$ . Also consider an  $(m - 1)$ -processor system comprising of processors  $p_1, p_2, \dots, p_{m-1}$  following the same sequence of load distribution as the above-mentioned  $m$ -processor system. Let the value of  $\alpha_1$  for these two systems be  $\alpha_1(m)$  and  $\alpha_1(m - 1)$ , respectively. In this situation,

$\alpha_1(m) < \alpha_1(m - 1)$  only when  $X(m) < 1$ . Or in other words, the processing time for the  $m$ -processor system is less than the processing time for the  $(m - 1)$ -processor system, only when  $X(m) < 1$ .

PROOF. The value of  $\alpha_1(m)$  and  $\alpha_1(m - 1)$  are as follows:

$$\alpha_1(m) = \frac{M_1(m) + Z(m)}{Y(m)}, \tag{12}$$

$$\alpha_1(m - 1) = \frac{M_1(m - 1) + Z(m - 1)}{Y(m - 1)}. \tag{13}$$

We have to obtain the condition under which  $\alpha_1(m) - \alpha_1(m - 1) < 0$ . Or in other words,

$$\frac{1}{D} [(M_1(m) + Z(m))Y(m - 1) - (M_1(m - 1) + Z(m - 1))Y(m)] < 0, \tag{14}$$

where  $D = Y(m)Y(m - 1)$ . This above expression reduces to

$$(Z(m)Y(m - 1) - Z(m - 1)Y(m)) < (M_1(m - 1)Y(m) - M_1(m)Y(m - 1)). \tag{15}$$

This can be further simplified as

$$\beta_1 + \beta_2 Y(2) + \dots + \beta_{m-1} Y(m - 1) < 1. \tag{16}$$

This above equation is the same as  $X(m) < 1$ . Hence,  $\alpha_1(m) < \alpha_1(m - 1)$  only when  $X(m) < 1$ . In the earlier study, it is shown that beyond this optimal number of processors  $m^*$ , an optimal solution ceases to exist. This is because  $X(m^* + k) > 1$ , for  $k = 1, 2, \dots$ . The reason for this is that some of the load fractions will be negative. In our approach, this fact is obtained as an increase in the processing time.

LEMMA 2. Consider an  $(m + 1)$ -processor system with a fixed sequence of load distribution as  $p_1, p_2, \dots, p_{m+1}$ . Also consider an  $m$ -processor system comprising of processors  $p_1, p_2, \dots, p_m$ , following the same load distribution as the above-mentioned  $(m + 1)$ -processor system. Let the value of  $\alpha_1$  for these two systems be  $\alpha_1(m + 1)$  and  $\alpha_1(m)$ , respectively. In this situation,  $\alpha_1(m) < \alpha_1(m + 1)$  only when  $X(m + 1) > 1$ . Or in other words, the processing time for the  $(m + 1)$ -processor system is more than the processing time of the  $m$ -processor system only when  $X(m + 1) > 1$ .

PROOF. The value of  $\alpha_1(m + 1)$  and  $\alpha_1(m)$  are as follows:

$$\alpha_1(m + 1) = \frac{M_1(m + 1) + z(m + 1)}{Y(m + 1)}, \tag{17}$$

$$\alpha_1(m) = \frac{M_1(m) + Z(m)}{Y(m)}. \tag{18}$$

We have to obtain the condition under which  $\alpha_1(m) - \alpha_1(m + 1) < 0$ . Or in other words,

$$\frac{1}{D} [\{M_1(m) + Z(m)\}Y(m + 1) - \{M_1(m + 1) + Z(m + 1)\}Y(m)] < 0, \tag{19}$$

where  $D = Y(m)Y(m + 1)$ . The above expression reduces to

$$\{Z(m)Y(m + 1) - Z(m + 1)Y(m)\} \leq \{M_1(m + 1)Y(m) - M_1(m)Y(m + 1)\}. \tag{20}$$

This can be further simplified as

$$\beta_1 + \beta_2 Y(2) + \dots + \beta_m Y(m) > 1. \tag{21}$$

This condition is the same as  $X(m+1) > 1$ . Hence,  $\alpha_1(m) < \alpha_1(m+1)$  only when  $X(m+1) > 1$ . From the above two lemmas, we can see that our closed-form expression for  $\alpha_1$  (and hence, the processing time) has a minimum for an optimal number of processor  $m^*$  such that

$$\begin{aligned} \alpha_1(m^*) &< \alpha_1(m^* - 1), \\ \alpha_1(m^*) &< \alpha_1(m^* + 1). \end{aligned} \tag{22}$$

The processing time will decrease with increase in processors up to  $m^*$ , and then the processing time is increasing with additional processors. Note that the necessary and sufficient condition given in [18] for the existence of an optimal processing time is satisfied in our approach. So the load fraction assigned to processors in our approach will be the same as the load fractions assigned to the processors in the earlier approach. Hence, we can say that  $m^*$  is the optimal number of processors only when  $\alpha_1(m^*) < \alpha_1(m^* - 1)$  and  $\alpha_1(m^*) < \alpha_1(m^* + 1)$ .

### Homogeneous System

As a special case, for a homogeneous system  $w_i = w$ , and hence,  $E_i = E$ , for  $i = 1, 2, \dots, m$ , we will show the condition on  $\beta$ , under which  $m$  is the optimal number of processors. For this, it is sufficient to consider the value of  $\alpha_1$ .

$$\alpha_1(m - 1) = \frac{f^{m-2} + \beta(1 + 2f + 3f^2 + \dots + (m - 2)f^{m-3})}{1 + f + f^2 + \dots + f^{m-2}}, \tag{23}$$

$$\alpha_1(m) = \frac{f^{m-1} + \beta(1 + 2f + 3f^2 + \dots + (m - 1)f^{m-2})}{1 + f + f^2 + \dots + f^{m-1}}, \tag{24}$$

$$\alpha_1(m + 1) = \frac{f^m + \beta(1 + 2f + 3f^2 + \dots + mf^{m-1})}{1 + f + f^2 + \dots + f^m}. \tag{25}$$

First, we will obtain the condition on  $\beta$  for which  $\alpha_1(m) < \alpha_1(m - 1)$ . Or in other words,

$$\frac{1}{D} \left[ \begin{aligned} &\{f^{m-1} + \beta(1 + 2f + 3f^2 + \dots + (m - 1)f^{m-2})\} (1 + f + f^2 + \dots + f^{m-2}) \\ &- \{f^{m-2} + \beta(1 + 2f + \dots + (m - 2)f^{m-3})\} (1 + f + f^2 + \dots + f^{m-1}) \end{aligned} \right] < 0, \tag{26}$$

where  $D$  is the product of denominators of  $\alpha_1(m)$  and  $\alpha_1(m - 1)$ . Following the same manner as in Lemma 1, this reduces to

$$\beta [(m - 1) + (m - 2)f + (m - 3)f^2 + \dots + f^{m-2}] < 1. \tag{27}$$

Now we will prove the condition on  $\beta$  for which  $\alpha_1(m) < \alpha_1(m + 1)$ , i.e.,

$$\frac{1}{D} \left[ \begin{aligned} &\{f^{m-1} + \beta(1 + 2f + 3f^2 + \dots + (m - 1)f^{m-2})\} (1 + f + f^2 + \dots + f^m) \\ &- \{f^m + \beta(1 + 2f + 3f^2 + \dots + mf^{m-1})\} (1 + f + f^2 + \dots + f^{m-1}) \end{aligned} \right] < 0. \tag{28}$$

Here  $D$  is the product of the denominator of  $\alpha_1(m)$  and  $\alpha_1(m + 1)$ . This expression reduces to

$$\beta (m + (m - 1)f + \dots + f^{m-1}) > 1. \tag{29}$$

From the above two equations, we can say  $m$  is the optimal number of processors only when

$$\beta < \frac{1}{(m - 1) + (m - 2)f + \dots + f^{m-2}} \tag{30}$$

and

$$\beta > \frac{1}{m + (m - 1)f + \dots + f^{m-1}}. \tag{31}$$

**Numerical Example**

We now present the numerical results obtained using the speed parameters given in [18]. In our approach, also the processing time is given by

$$T(\alpha, m) = \alpha_1(E_1 + C) + \theta_{cm} + \theta_{cp}, \tag{32}$$

as given in [18]. In our approach, it is sufficient to consider the behaviour of  $\alpha_1$ , to study the behaviour of the processing time. We know that

$$\alpha_1(m) = \frac{M_1(m) + Z(m)}{Y(m)}. \tag{33}$$

It can be seen that  $\alpha_1(m)$  has two components:

- (i)  $M_1(m)/Y(m)$ —component of  $\alpha_1(m)$  without overheads;
- (ii)  $Z(m)/Y(m)$ —component of  $\alpha_1(m)$  due to overheads.

We know that  $M_1(m)/Y(m)$  decreases with increasing  $m$  and  $Z(m)/Y(m)$  increases with increasing  $m$ . Similarly, the processing time also has two components. With the speed parameters, given in [18], the processing time obtained for  $C = 0.4$  and  $C = 0.2$  are given in Table 1. From Table 1, we can see for  $C = 0.4$ , the processing time decreases up to the optimal number of processors ( $m^* = 6$ ) and then starts increasing. For the case  $C = 0.2$ , the processing time decreases up to the optimal number of processors ( $m^* = 10$ ), and then increases. As expected, the optimal number of processors is the same as obtained in [18]. In Figure 3, the behaviour of processing time with the number of processors is shown for  $C = 0.2$ . In Figure 3, the component of processing time without the overhead, the processing time component because of the overhead, and the total of the two components are shown. Because of the numerical values  $E_i$  and  $\theta_{cm}$ , the increase in the overhead components is very small with the increase in processors, and hence, the

Table 1. Processing time with number of processors.

Number of Processors	Processing Time $C = 0.4$	Processing Time $C = 0.2$
1	0.7370000	0.521000
2	0.4884439	0.307428
3	0.4345488	0.244888
4	0.4291981	0.237428
5	0.4275042	0.234157
6	0.426954*	0.232272
7	0.4269693	0.231183
8		0.230253
9		0.230021
10		0.230017*
11		0.230151
12		0.230462
13		0.231102
14		0.231934
15		0.233165

Speed parameters form [18].

$\theta_{cp}$	$\theta_{cm}$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$	$E_8$	$E_9$	$E_{10}$	$E_{11}$	$E_{12}$	$E_{13}$	$E_{14}$	$E_{15}$
0.02	0.001	0.3	0.2	0.1	0.4	0.6	0.7	0.8	0.5	0.9	1.1	1.3	1.0	0.6	0.5	0.3



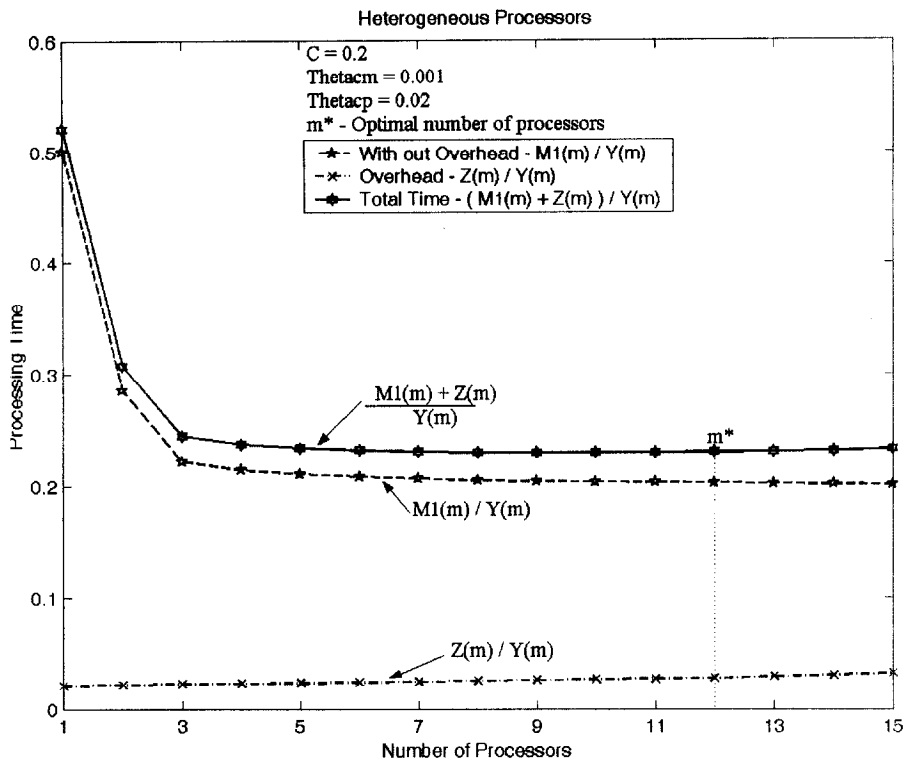


Figure 3. Variation of processing time with number of processors (heterogeneous case).

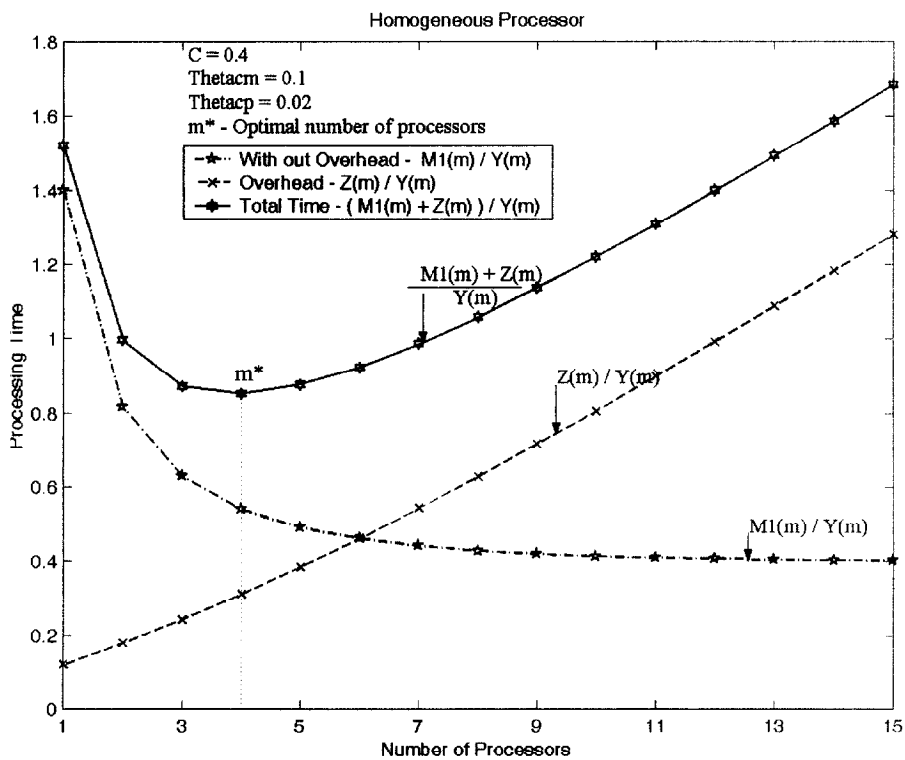


Figure 4. Variation of processing time with number of processors (homogeneous case).

decrease and increase in the processing time before and after the optimal number of processors is small. Figure 4 presents the processing time results for the homogeneous network with numerical values  $E_i = 1$ , for  $i = 1, 2, \dots, m$ ,  $C = 0.4$ ,  $\theta_{cm} = 0.1$ , and  $\theta_{cp} = 0.02$ . In this figure, we can see that the increase in overhead components is not small (as in the heterogeneous case), and hence, the behaviour of processing time before and after the optimal number of processors is more clear.

It is important to note here the following: we are not using this closed-form expression to obtain the load fraction beyond the optimal number of processors  $m^*$ . It is mentioned in [18] that, beyond this  $m^*$ , the optimal solution ceases to exist. This is because some of the load fractions will be negative. This fact that some of the load fractions are negative is reflected in our approach as an increase in the values of  $\alpha_1$ .

#### 4. CONCEPT OF SEQUENCING

The advantage of our closed-form expression is that this can be directly used to obtain the optimal sequence of load distribution. For the sake of clarity, we will first illustrate the optimal sequence for the case with  $m = 3$  and then generalize the result. We also assume that  $X(3) < 1$ . The value of  $\alpha_1$ , for a given sequence of load distribution, is

$$\alpha_1 = \frac{f_3 f_2 + \beta_1 (1 + f_3) + \beta_2 f_2}{1 + f_3 + f_3 f_2}. \quad (34)$$

We will rewrite this above  $\alpha_1$  expression in terms of  $E_i$  ( $i = 1, 2, 3$ ) and  $\theta_{cm}$  as

$$\alpha_1 = \frac{(E_3 + C)(E_2 + C) + \theta_{cm}(2E_2 + E_3 + 2C)}{E_2 E_1 + E_1(E_3 + C) + (E_3 + C)(E_2 + C)}. \quad (35)$$

Note here in the above expression, the sequence of load distribution is  $(p_1, p_2, p_3)$ , i.e., the BCU first sends the load fraction to processor  $p_1$  (speed  $E_1$ ), next to processor  $p_2$  (speed  $E_2$ ), and last, to processor  $p_3$  (speed  $E_3$ ). Let the BCU change the sequence of load distribution to  $(p_1, p_3, p_2)$ , i.e., first send the load fraction to processor  $p_1$  (speed  $E_1$ ), next to processor  $p_3$  (speed  $E_3$ ), and last, to processor  $p_2$  (speed  $E_2$ ). We will denote the value of  $\alpha_1$  for this sequence as  $\alpha'_1$ . Note that  $\alpha'_1$  can be obtained by interchanging  $E_2$  and  $E_3$  in the earlier expression and is obtained as

$$\alpha'_1 = \frac{(E_2 + C)(E_3 + C) + \theta_{cm}(2E_3 + E_2 + 2C)}{E_3 E_1 + E_1(E_2 + C) + (E_2 + C)(E_3 + C)}. \quad (36)$$

We have to find the condition for which  $\alpha_1 \leq \alpha'_1$ . The denominators of the  $\alpha_1$  and  $\alpha'_1$  are the same. Also, the first term in the numerator of  $\alpha_1$  and  $\alpha'_1$  are the same. Hence,

$$\alpha_1 - \alpha'_1 = \frac{\theta_{cm}}{D} \{2E_2 + E_3 + 2C - 2E_3 - E_2 - 2C\} = \frac{\theta_{cm}(E_2 - E_3)}{D}, \quad (37)$$

where  $D$  is the denominator of  $\alpha'_1$  or  $\alpha_1$ . From this, we can say that the processing time for the sequence  $(p_1, p_2, p_3)$  is less than or equal to the processing time for the sequence  $(p_1, p_3, p_2)$  only when  $E_2$  is less than or equal to  $E_3$ .

#### Generalization

For  $m$  processors, consider the BCU distribute the load fraction to the processors in the following sequence:  $(p_1, p_2, p_3, \dots, p_i, p_{i+1}, \dots, p_m)$ . Let  $X(m) < 1$ . The value of  $\alpha_1$  for this sequence denoted by  $\alpha_1(m)$  is

$$\alpha_1(m) = \frac{M_1(m) + Z(m)}{Y(m)}. \quad (38)$$

Consider another sequence of load distribution by the BCU to the processors as  $(p_1, p_2, p_3, \dots, p_{i+1}, p_i, \dots, p_m)$ . Let  $X(m) < 1$ . The value of  $\alpha_1$  and for this load distribution denoted as  $\alpha'_1(m)$  is

$$\alpha'_1(m) = \frac{M'_1(m) + Z'(m)}{Y'(m)}. \tag{39}$$

$\alpha'_1(m)$  can be obtained by interchanging  $E_i$  and  $E_{i+1}$  in  $\alpha_1(m)$ . Because of this interchange, only  $f_{i+2}, f_{i+1}, f_i, \beta_i, \beta_{i+1}$  will change. The other quantities will not change. Note that, because of this interchange,  $M_1(m)$  and  $M'_1(m)$  will not change. Also because of this interchange,  $Y(m)$  and  $Y'(m)$  also will not change, i.e.,

$$\begin{aligned} M_1(m) &= M'_1(m), \\ Y(m) &= Y'(m). \end{aligned} \tag{40}$$

This above fact can be verified from the optimal sequence Lemma 7.1 given in [8] for a single-level tree network with  $\tau = 1$ . In Lemma 7.1,  $\tau = 1$  implies that link speeds are the same as in the case for the bus network, and this is also shown to be true in Theorem 7.2 for a single-level tree network given in [8].

Now we have to find the condition for which  $\alpha_1(m) \leq \alpha'_1(m)$ . This is the same as to find the condition for which  $Z(m) \leq Z'(m)$ . We know that  $Z(m)$  and  $Z'(m)$  are functions of  $\beta_1, \beta_2, \dots, \beta_{m-1}$ . So we consider this term-by-term.  $\beta_1$  terms in  $Z(m)$  are

$$\frac{\theta_{cm}}{E_1} \left[ \begin{aligned} &1 + f_m + f_m f_{m-1} + \dots + f_m f_{m-1} \dots f_{i+2} \\ &+ f_m f_{m-1} \dots f_{i+2} f_{i+1} + f_m f_{m-1} \dots f_{i+2} f_{i+1} f_i + \dots + f_m f_{m-1} \dots f_4 f_3 \end{aligned} \right]. \tag{41}$$

When we interchange  $E_i$  and  $E_{i+1}$  in the above expression only  $f_{i+2}, f_{i+1}, f_i$  will change. The changed values are denoted as  $g_{i+2}, g_{i+1}$ , and  $g_i$  defined as follows:

$$\begin{aligned} g_{i+2} &= \frac{E_{i+2} + C}{E_i}, \\ g_{i+1} &= \frac{E_i + C}{E_{i+1}}, \\ g_i &= \frac{E_{i+1} + C}{E_{i-1}}. \end{aligned} \tag{42}$$

$\beta_1$  terms in  $Z'(m)$  are obtained by replacing  $f_{i+2}$  by  $g_{i+2}$ ,  $f_{i+1}$  by  $g_{i+1}$ , and  $f_i$  by  $g_i$  in  $\beta_1$  terms of  $Z(m)$  and is obtained as

$$\frac{\theta_{cm}}{E_1} \left[ \begin{aligned} &1 + f_m + f_m f_{m-1} + \dots + f_m f_{m-1} \dots g_{i+2} + \\ &+ f_m f_{m-1} \dots g_{i+2} g_{i+1} + f_m f_{m-1} \dots g_{i+2} g_{i+1} g_i + \dots + f_m f_{m-1} \dots g_{i+2} g_{i+1} g_i \dots f_3 \end{aligned} \right]. \tag{43}$$

Note that  $f_{i+2} f_{i+1} f_i = g_{i+2} g_{i+1} g_i$ . Hence, from the  $\beta_1$  terms in  $Z(m)$  and  $Z'(m)$ , we get the contribution of  $\beta_1$  terms in  $Z(m) - Z'(m)$  as

$$\frac{\theta_{cm}}{E_1} [f_m f_{m-1} \dots f_{i+3} \{f_{i+2} + f_{i+2} f_{i+3} - g_{i+2} + g_{i+2} g_{i+3}\}]. \tag{44}$$

The value of  $\beta_1$  terms in  $Z(m) - Z'(m)$  is zero since

$$f_{i+2} + f_{i+2} f_{i+3} - g_{i+2} + g_{i+2} g_{i+3} = 0. \tag{45}$$

In a similar way, it can be easily shown that all  $\beta_j$  terms ( $j = 1, 2, \dots, m - 1$ , and  $j \neq i$ ) vanishes, except  $\beta_i$  terms in the expression  $Z(m) - Z'(m)$ . Or in other words, only  $\beta_i$  terms in  $Z(m)$  and  $Z'(m)$  will have a nonzero value in  $Z(m) - Z'(m)$ . Hence,  $Z(m) - Z'(m)$  is obtained as

$$Z(m) - Z'(m) = K(E_i - E_{i+1}), \tag{46}$$

where

$$K = \frac{E_m E_{m-1} \dots E_{i+2} (E_{i-1} + c)(E_{i-2} + C) \dots (E_2 + c)}{E_m E_{m-1} \dots E_2} \theta_{cm}.$$

Hence,  $\alpha_1(m) \leq \alpha'_1(m)$  only when  $E_i \leq E_{i+1}$ . Based on this generalization, we state the following lemma.

LEMMA 3. The processing time for the sequence  $(p_1, p_2, \dots, p_i, p_{i+1}, \dots, p_m)$  is less than or equal to the processing time for the sequence  $(p_1, p_2, \dots, p_{i+1}, p_i, \dots, p_m)$  only when  $E_i \leq E_{i+1}$ .

The concept of sequencing proposes a method by which the minimum processing time can be achieved. However, we have not included the first processor in the concept of sequencing, i.e., in the interchange argument  $i = 2, 3, \dots, m - 1$ . Now, we will prove the speed condition on the first processor. For this purpose, we consider a bus network with only two processors,  $p_1$  (speed  $E_1$ ) and  $p_2$  (speed  $E_2$ ).

CASE (i). SEQUENCE OF LOAD DISTRIBUTION  $(p_1, p_2)$ . Let  $T(\alpha, 2)$  be the processing time for this sequence of load distribution and is obtained as

$$T(\alpha, 2) = \frac{E_2 + C + \theta_{cm}}{E_2 + E_1 + C} (E_1 + C) + \theta_{cp} + \theta_{cm}. \quad (47)$$

CASE (ii). SEQUENCE OF LOAD DISTRIBUTION  $(p_2, p_1)$ . Let  $T(\alpha', 2)$  be the processing time for this sequence of load distribution and is obtained as

$$T(\alpha', 2) = \frac{E_1 + C + \theta_{cm}}{E_2 + E_1 + C} (E_2 + C) + \theta_{cp} + \theta_{cm}. \quad (48)$$

The denominators of  $T(\alpha', 2)$  and  $T(\alpha, 2)$  are the same. We obtain the condition on  $T(\alpha, 2) - T(\alpha', 2)$  as

$$T(\alpha, 2) - T(\alpha', 2) = \frac{1}{D} \{ (E_2 + C + \theta_{cm})(E_1 + C) - (E_1 + C + \theta_{cm})(E_2 + C) \}, \quad (49)$$

where  $D$  is the denominator of  $T(\alpha', 2)$  (or  $T(\alpha, 2)$ ).

This reduces to

$$T(\alpha, 2) - T(\alpha', 2) = \frac{\theta_{cm}}{E_2 + E_1 + C} (E_1 - E_2). \quad (50)$$

Hence,  $T(\alpha, 2) \leq T(\alpha', 2)$  only when  $E_1 \leq E_2$ . From here, we can say the first processor should be the fastest. Note that, to find the speed condition of the first processor, we have to use the processing time expression. For the speed condition of other processors, it is sufficient to consider the value of the  $\alpha_1$  expression rather than the processing time expression. Though we have chosen only two processors to prove the condition on speed of the first processor, for an  $m$ -processor system this can be easily proved in a similar fashion, as done for a single-level tree network in Lemma 7.3 given in [8].

In the earlier study [18], the fast sequence is defined as the sequence  $(p_1, p_2, \dots, p_i, p_{i+1}, \dots, p_m)$  such that  $E_i < E_{i+1}$  for all  $i = 1, 2, \dots, m - 1$ . For an  $m$ -processor system,  $m!$  different load distribution sequences are possible. It is possible in this analysis to have a nonoptimal sequence of load distribution and use additional processors. For example, let the optimal number of processors for an  $m$ -processor system, using an optimal sequence, be  $m^*$ , and the value of  $\alpha_1$  for this is  $\alpha_1(m^*)$ .

Let the optimal number of processors for the same  $m$ -processor system using a nonoptimal sequence be  $m^* + k$ , and the value of  $\alpha_1$  for this is  $\alpha'_1(m^* + k)$ . Here, because the sequence is nonoptimal, we can use more processors. We have to prove that  $\alpha_1(m^*) < \alpha'_1(m^* + k)$ , i.e., we have to prove that the processing time with optimal sequence and optimal number of processors is less than the processing time with nonoptimal sequence and the corresponding optimal number of processors (for this nonoptimal sequence). By rearranging, the nonoptimal sequence, using the sequencing analysis, we can obtain the optimal sequence. Let the value of  $\alpha_1$  obtained after rearrangement be  $\alpha_1(m^* + k)$ . Based on the sequencing analysis, we know that the value of  $\alpha_1$  with an optimal sequence is less than the value of  $\alpha_1$  with a nonoptimal sequence, i.e.,  $\alpha_1(m^* + k) < \alpha'_1(m^* + k)$ .

Now, we know that  $\alpha_1(m^*)$  and  $\alpha_1(m^* + k)$  are obtained using an optimal sequence. From Lemmas 1 and 2, we know that for any given sequence of load distribution  $\alpha_1(m^*) < \alpha_1(m^* + k)$ , and hence,  $\alpha_1(m^*) < \alpha'_1(m^* + k)$ . Based on the above analysis, we can state the following lemma.

LEMMA 4. *The optimal processing time is the processing time obtained using an optimal sequence of load distribution with an optimal number of processors.*

## CONCLUSIONS

The effect of start-up in scheduling divisible loads on a bus network is considered and an alternate approach is presented to obtain the processing time. In the earlier approach [18], first the value of  $\alpha_m$  is obtained (using the necessary and sufficient condition  $X(m) < 1$ ), and then the value of  $\alpha_1$  and the processing time are obtained. In our approach presented in this paper, a direct closed-form expression for the value of  $\alpha_1$ , and hence, the processing time is presented. It is also proved that the optimal number of processors obtained, using this closed-form expression, satisfies the necessary and sufficient conditions presented in [18]. Using this closed-form expression, we prove important results in sequencing. It is proved analytically, in this paper that, for a bus network sharing a divisible load with start-up delays, the optimal processing time is obtained using optimal sequence of load distribution with optimal number of processors.

## REFERENCES

1. S.H. Bokhari, *Assignment Problems in Parallel and Distributed Computing*, Kluwer Academic, Boston, MA, (1987).
2. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, New York, (1979).
3. Y.C. Cheng and T.G. Robertazzi, Distributed computation with communication delays, *IEEE Trans., Aerospace and Electronics Systems* **24** (6), 700–712, (1988).
4. Y.C. Cheng and T.G. Robertazzi, Distributed computation for a tree network with communication delays, *IEEE Trans., Aerospace and Electronic Systems* **26** (3), 511–516, (1990).
5. S. Bataineh and T.G. Robertazzi, Bus oriented load sharing for a network of sensor driven processors. *IEEE Trans., System, Man, Cybernetics* **21** (5), 1202–1205, (1991).
6. T.G. Robertazzi, Processor equivalent for a linear daisy chain of load sharing processors, *IEEE Trans., Aerospace and Electronic Systems* **29** (4), 1216–1221, (1993).
7. J. Sohn and T.G. Robertazzi, Optimal divisible job load sharing on bus network, *IEEE Trans., Aerospace and Electronic Systems* **32** (1), 34–40, (1996).
8. V. Bharadwaj, D. Ghose, V. Mani and T.G. Robertazzi, *Scheduling, Divisible Loads in Parallel and Distributed Systems*, IEEE Computer Society Press, Los Alamitos, CA, (1996).
9. V. Bharadwaj, D. Ghose and V. Mani, Optimal sequencing and arrangement in distributed single-level networks with communication delays, *IEEE Trans. Parallel and Distributed Systems* **5** (9), 968–976, (1994).
10. H.J. Kim, G.-I. Jee and J.G. Lee, Optimal load distribution for tree network processors, *IEEE Trans., Aerospace and Electronics Systems* **32** (2), 607–612, (1996).
11. V. Mani and D. Ghose, Distributed computation in linear networks: Closed-form solutions, *IEEE Trans., Aerospace and Electronics Systems* **30** (2), 471–483, (1994).
12. D. Ghose and V. Mani, Distributed computation with communication delays: Asymptotic performance analysis, *J. Parallel and Distributed Computing* **23** (3), 293–305, (1994).
13. D. Ghose and H.J. Kim, Load partitioning and trade-off study for large matrix vector computations in multicast bus networks with communication delays, *J. Parallel and Distributed Computation* **55** (1), 32–59, (1998).
14. D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, (1989).
15. G.D. Barlas, Collection-aware optimum sequencing of operations and closed-form solutions for the distribution of a divisible load on arbitrary processors trees, *IEEE Trans. Parallel and Distributed Systems* **9** (5), 429–441, (1998).
16. J. Blazewicz and M. Drozdowski, Distributed processing of divisible loads with communication startup cost, *Discrete Applied Math.* **76** (1–3), (1997).
17. M. Drozdowski, *Selected Problems of Scheduling Tasks in Multiprocessor Computer Systems, No. 321*. Wydawnictwa Politechniki Pozanskiej, Pozan, Poland, (1997).
18. V. Bharadwaj, X. Li and C.C. Ko, On the influence of start-up costs in scheduling divisible loads on bus networks, *IEEE Trans. on Parallel and Distributed Systems* **11** (12), 1288–1305, (2000).