STATE UNIVERSITY OF NEW YORK AT

STONY BROOK

CEAS TECHNICAL REPORT 644

Optimal Load Sharing for a Divisible Job on a Bus Network

J. Sohn & T.G. Robertazzi

October 22, 1992

Optimal Load Sharing for a Divisible Job on a Bus Network

Jeeho Sohn and Thomas G. Robertazzi, *Senior Member, IEEE*

Dept. of Electrical Engineering,

SUNY at Stony Brook,

Stony Brook, N.Y. 11794

Abstract

Optimal load allocation for load sharing a divisible job over N processors interconnected in bus-oriented network is considered. The processors are equipped with front-end processors. It is analytically proved, for the first time, that a minimal solution time is achieved when the computation by each processor finishes at the same time. Closed form solutions for the minimum finish time and the optimal data allocation for each processor are obtained.

I INTRODUCTION

In recent years, there has been of great interest in distributed sensor networks [1]. In distributed sensor networks, measurements are made by spatially distinct sensors. The data is, then, broadcast to a site where the spatially disparate readings are fused so that meaningful decisions can be made regarding these measurements. One major issue for distributed sensor networks is the trade-off between communication and computation [2]. That is, the decision of how much time should be spent to communicate and how much time should be spent to process (compute) the measurements becomes an important problem.

Related to the distributed sensor network problem are a number of papers which deal with scheduling and load sharing in multiprocessors [3, 4]. However most work assumes that a job can be assigned to a single processor. Only recently has there been interest in multiprocessor scheduling with jobs that need to be assigned to more than one processor [5, 6, 7].

Recently there has been work on a load sharing problem involving a *divisible* job. A divisible job is a job that can be arbitrarily portioned in a linear fashion among a number of processors. Applications include the processing of very long data files as in signal and image processing and Kalman filtering.

In [8], recursive expressions for calculating the optimal load allocation for linear daisy chains of processors were presented. This is based on the simplifying premise that for an optimal allocation of load, all processors must stop processing at the same time. Intuitively, this is because otherwise some processors would be idle while others were still busy. Analogous solutions have been developed for tree networks [9] and bus networks [10, 11]. The

equivalence of first distributing load either to the left or to the right from a point in the interior of a linear daisy chain is demonstrated in [17]. Optimal sequences of load distribution in tree networks are described in [16, 18, 19, 20]. Closed form solutions for homogeneous bus and tree networks appear in [12]. Asymptotic solutions for systems with infinite number of processors appear in [13, 21].

In [14], the concept of processor equivalence was used to prove that the optimal allocation of load for a linear daisy chain of processors involves having all the processors finish computation at the same time. However, until now there has been no analytic proof available that the same type of solution is optimal for N processors interconnected through a bus type channel. What has been available, aside from intuition, is a proof for the two processors case (N=2), and computational results consistent with the *all processors stop at the same time* premise [10, 11]. In this paper an analytic proof is presented for the case of a bus network where each processor has a front-end processor for communications off loading. The proof shows that for the general N processors case the minimum time solution occurs when all processors finish computation at the same time. A by product of the proof are closed form solutions for the optimal load allocation when processor speeds are heterogeneous (see [12] for the case of homogeneous processor speeds).

This paper is organized as follows. In section II the proof is presented in a recursive fashion. Generalizations of the proof to alternative architectures are discussed in section III. The conclusion appears in section IV.

II Optimal Division of Processing Load: Bus Network with Front-End Processors

Consider a bus network that consists of N processors as shown in Fig. 1. Each processor is equipped with front-end processors for communications off-loading. That is, the processors can communicate and compute at the same time. Any single processor can receive a burst of measurement data and distributes the processing load among the processors to obtain the benefits of parallel processing. Each processor may have a different computing speed.

The following notation will be used throughout this paper:

α_i : The fraction of the entire processing load that is assigned to ith processor.

Z : A constant that is inversely proportional to channel speed of bus.

w_i : A constant that is inversely proportional to the computing speed of ith processor.

T_{cm} : The time that it takes to transmit the entire set of measurement data over the channel when $Z = 1$.

T_{cp} : The time that it takes for ith processor to process (compute) the entire load when $w_i = 1$.

T_i : The finish time of the ith processor's computation, assuming the load is delivered to the originating processor at time zero.

Fig. 2 shows the timing diagram for the processors which are equipped with front-end processors in a bus network. At time $t = 0$, the originating processor 1 transmits the second

fraction of processing load to the processor 2 and starts computing the first fraction of processing load simultaneously. When the transmission of the second fraction of processing load is finished, processor 1 then transmits the third fraction of processing load to processor 3. In the mean time, processor 2 starts computing the received processing load. The process then continues on in the natural way. The equations that represent the finish time of each processor are given by

$$T_1 = \alpha_1 w_1 T_{cp} \tag{1}$$

$$T_2 = \alpha_2 Z T_{cm} + \alpha_2 w_2 T_{cp} \tag{2}$$

$$T_3 = (\alpha_2 + \alpha_3) Z T_{cm} + \alpha_3 w_3 T_{cp} \tag{3}$$

$$\vdots$$

$$T_i = (\alpha_2 + \alpha_3 + \cdots + \alpha_i) Z T_{cm} + \alpha_i w_i T_{cp} \tag{4}$$

$$\vdots$$

$$T_N = (\alpha_2 + \alpha_3 + \cdots + \alpha_N) Z T_{cm} + \alpha_N w_N T_{cp} \tag{5}$$

The fractions of total measurement load should sum to one.

$$\alpha_1 + \alpha_2 + \cdots + \alpha_i + \cdots + \alpha_N = 1 \tag{6}$$

The necessary conditions to achieve the minimum solution time will be examined through the following subsections in a recursive manner.

A *Consideration of* T_1 *and* T_2

First consider T_1 and T_2, the finish times of computation of processor 1 and processor 2, respectively. On the other hand, the rest of processing finish times, $T_3, T_4, \ldots, T_N$, will not be considered yet and will be assumed to have arbitrary values. That is, the fractions, $\alpha_3, \alpha_4, \ldots, \alpha_N$, are assumed to have arbitrary constant values. They will be considered in the following subsections with some results obtained in this subsection.

Let C_2 be the sum of $\alpha_3, \alpha_4, \ldots, \alpha_N$

$$C_2 = \alpha_3 + \alpha_4 + \cdots + \alpha_N \tag{7}$$

where C_2 is a constant. Then

$$\begin{aligned} \alpha_2 &= 1 - (\alpha_3 + \alpha_4 + \cdots + \alpha_N) - \alpha_1 \\ &= (1 - C_2) - \alpha_1 \end{aligned} \tag{8}$$

where α_1 has its maximum value when $\alpha_2 = 0$:

$$0 \leq \alpha_1 \leq 1 - C_2 \tag{9}$$

Then, T_1 and T_2 can be represented as follows:

$$T_1 = \alpha_1 w_1 T_{cp} \tag{10}$$

$$T_2 = \alpha_2 Z T_{cm} + \alpha_2 w_2 T_{cp} \tag{11}$$

$$= (Z T_{cm} + w_2 T_{cp})[(1 - C_2) - \alpha_1] \tag{12}$$

The optimal processing time is the time that minimizes $\max(T_1, T_2)$. As shown in Fig. 3, the optimal processing time is achieved at the crossover point of the two lines where $T_1 = T_2$.

From Eq.(10) and Eq.(11), α_2 can be expressed as a function of α_1 since $T_1 = T_2$:

$$\begin{aligned} \alpha_2 &= \frac{w_1 T_{cp}}{Z T_{cm} + w_2 T_{cp}} \alpha_1 \\ &= k_1 \alpha_1 \end{aligned} \tag{13}$$

where $k_1 = \frac{\alpha_2}{\alpha_1} = \frac{w_1 T_{cp}}{Z T_{cm} + w_2 T_{cp}}$. Note that a simple minimum is always present in this diagram.

B *Consideration of T_1, T_2 and T_3*

This subsection will examine the optimal processing time when T_1, T_2 and T_3 are considered. This consideration will include some information which was obtained from the previous subsection, namely $T_1 = T_2$. We will assume $\alpha_4, \alpha_5, \ldots, \alpha_N$ to have arbitrary constant values.

Let C_3 be the sum of $\alpha_4, \alpha_5, \ldots, \alpha_N$.

$$C_3 = \alpha_4 + \alpha_5 + \cdots + \alpha_N \tag{14}$$

where C_3 is a constant. Then

$$\begin{aligned} \alpha_3 &= 1 - (\alpha_4 + \alpha_5 + \cdots + \alpha_N) - \alpha_1 - \alpha_2 \\ &= (1 - C_3) - (\frac{1}{k_1} + 1)\alpha_2 \end{aligned} \tag{15}$$

since $\alpha_1 = \frac{\alpha_2}{k_1}$. Now α_2 has its maximum value when $\alpha_3 = 0$.

$$0 \leq \alpha_2 \leq (1 - C_3)\frac{k_1}{1 + k_1} \tag{16}$$

Then T_1, T_2 and T_3 can be represented as follows:

$$T_1 = T_2 = (ZT_{cm} + w_2T_{cp})\alpha_2 \tag{17}$$

$$T_3 = (\alpha_2 + \alpha_3)ZT_{cm} + \alpha_3 w_3 T_{cp} \tag{18}$$

$$= \alpha_2 ZT_{cm} + (ZT_{cm} + w_3T_{cp})[(1 - C_3) - (\frac{1}{k_1} + 1)\alpha_2] \tag{19}$$

$$= (1 - C_3)(ZT_{cm} + w_3T_{cp}) - [\frac{1}{k_1}ZT_{cm} + (\frac{1}{k_1} + 1)w_3T_{cp}]\alpha_2 \tag{20}$$

T_3 has its minimum value and $T_1 = T_2$ has its maximum value when α_2 reaches its maximum value, that is,

$$\min(T_3) = T_3\{\ \alpha_2 = (1 - C_3)\frac{k_1}{1 + k_1}\ \}$$

$$= (1 - C_3)\frac{k_1}{1 + k_1}ZT_{cm} \tag{21}$$

$$\max(T_1 = T_2) = T_2\{\ \alpha_2 = (1 - C_3)\frac{k_1}{1 + k_1}\ \}$$

$$= (1 - C_3)\frac{k_1}{1 + k_1}(ZT_{cm} + w_2T_{cp}) \tag{22}$$

Naturally, there exists a crossover point across the two lines since $(1 - C_3)\frac{k_1}{1+k_1}(ZT_{cm} + w_2T_{cp}) > (1 - C_3)\frac{k_1}{1+k_1}ZT_{cm}$. Then the optimal processing time is achieved at the crossover point of the two lines where $T_1 = T_2 = T_3$ as in Fig. 4.

From Eq.(17), Eq.(18) and Eq.(13), α_3 can be expressed as a function of α_2 and α_1 since $T_1 = T_2 = T_3$.

$$\alpha_3 = \frac{w_2T_{cp}}{ZT_{cm} + w_3T_{cp}}\alpha_2$$

$$= k_2\alpha_2$$

$$= k_2k_1 \cdot \alpha_1 \tag{23}$$

where $k_2 = \frac{\alpha_3}{\alpha_2} = \frac{w_2T_{cp}}{ZT_{cm}+w_3T_{cp}}$.

C *Consideration of* $T_1, T_2, \ldots, T_i$

Based on the results of the previous subsections, one can extend the proof to show that $T_1 = T_2 = \cdots = T_i$ achieves the minimal solution time. The assumption that $\alpha_{i+1}, \alpha_{i+2}, \ldots, \alpha_N$ have some arbitrary constant values will be also hold in this subsection.

Let C_i be the sum of $\alpha_{i+1}, \alpha_{i+2}, \ldots, \alpha_N$.

$$C_i = \alpha_{i+1} + \alpha_{i+2} + \cdots + \alpha_N \tag{24}$$

where C_i is a constant. Then

$$\begin{aligned} \alpha_2 + \alpha_3 + \cdots + \alpha_i &= 1 - (\alpha_{i+1} + \alpha_{i+2} + \cdots + \alpha_N) - \alpha_1 \\ &= (1 - C_i) - \frac{1}{k_1}\alpha_2 \end{aligned} \tag{25}$$

and

$$\begin{aligned} \alpha_i &= 1 - (\alpha_{i+1} + \alpha_{i+2} + \cdots + \alpha_N) - (\alpha_1 + \alpha_2 + \cdots + \alpha_{i-1}) \\ &= (1 - C_i) - (\frac{1}{k_1} + 1 + k_2 + k_3 k_2 + \cdots + k_{i-2} k_{i-3} \cdots k_3 k_2)\alpha_2 \end{aligned} \tag{26}$$

where α_2 has its maximum value when $\alpha_i = 0$:

$$0 \leq \alpha_2 \leq (1 - C_i)\frac{k_1}{1 + k_1 + k_1 k_2 + \cdots + k_1 k_2 \cdots k_{i-2}} \tag{27}$$

Then, $T_1, T_2, \ldots, T_i$ can be represented as follows:

$$T_1 = T_2 = \cdots = T_{i-1} = (ZT_{cm} + w_2 T_{cp})\alpha_2 \tag{28}$$

$$T_i = (\alpha_2 + \alpha_3 + \cdots + \alpha_i)ZT_{cm} + \alpha_i w_i T_{cp} \tag{29}$$

Using the representation of Eq.(25) and Eq.(26) and simplifying results in:

$$\begin{aligned} T_i &= [(1-C_i) - \frac{1}{k_1}\alpha_2]ZT_{cm} + [(1-C_i) - (\frac{1}{k_1} + 1 + k_2 + \cdots + k_{i-2}k_{i-3}\cdots k_2)\alpha_2]w_iT_{cp} \\ &= (1-C_i)(ZT_{cm} + w_iT_{cp}) - [\frac{1}{k_1}ZT_{cm} + (\frac{1}{k_1} + 1 + k_2 + \cdots + k_{i-2}k_{i-3}\cdots k_2)w_iT_{cp}]\alpha_2 \end{aligned} \tag{30}$$

T_i has its minimum value and $T_1 = T_2 = \cdots = T_{i-1}$ has its maximum value when α_2 reaches its maximum value, that is,

$$\begin{aligned} \min(T_i) &= T_i\{\alpha_2 = (1-C_i)\frac{k_1}{1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}\} \\ &= (1-C_i)\frac{k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}{1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}ZT_{cm} \end{aligned} \tag{31}$$

$$\begin{aligned} \max(T_1 = T_2 = \cdots = T_{i-1}) &= T_2\{\alpha_2 = (1-C_i)\frac{k_1}{1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}\} \\ &= (1-C_i)\frac{k_1}{1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}(ZT_{cm} + w_2T_{cp}) \end{aligned} \tag{32}$$

The following condition, based on the above, must be satisfied in order for a crossover point to exist between the two lines:

$$\begin{aligned} &(1-C_i)\frac{k_1}{1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}(ZT_{cm} + w_2T_{cp}) \\ &\quad > (1-C_i)\frac{k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}{1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2}}ZT_{cm} \end{aligned} \tag{33}$$

Proof: The above condition can be reduced as follows:

$$k_1(ZT_{cm} + w_2T_{cp}) > (k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{i-2})ZT_{cm} \tag{34}$$

$$w_2T_{cp} > k_2(1 + k_3 + k_3k_4 + \cdots + k_3k_4\cdots k_{i-2})ZT_{cm} \tag{35}$$

$$w_2 T_{cp} > \frac{\alpha_3}{\alpha_2}(1 + \frac{\alpha_4}{\alpha_3} + \frac{\alpha_4}{\alpha_3}\frac{\alpha_5}{\alpha_4} + \cdots + \frac{\alpha_4}{\alpha_3}\frac{\alpha_5}{\alpha_4}\cdots\frac{\alpha_{i-1}}{\alpha_{i-2}})ZT_{cm} \quad (36)$$

$$\alpha_2 w_2 T_{cp} > (\alpha_3 + \alpha_4 + \cdots + \alpha_{i-1})ZT_{cm} \quad (37)$$

Here $\alpha_2 w_2 T_{cp}$ is the processing time of processor 2. When processor 2 finishes its processing load at T_2 the other processors (1, 3, 4, ..., $i-1$) also finish their processing load at the same time which is implied in $T_1 = T_2 = \cdots = T_{i-1}$. Now $(\alpha_3 + \alpha_4 + \cdots + \alpha_{i-1})ZT_{cm}$ is the transmitting time from the third fraction ($\alpha_3 ZT_{cm}$) to $i-1$st fraction ($\alpha_{i-1} ZT_{cm}$). Suppose that the left hand side is less than the right hand side in Eq.(37). Then the finish time, $T_1 = T_2 = \cdots = T_{i-1}$, will be positioned somewhere before the end of $i-1$st transmitting period ($\alpha_{i-1} ZT_{cm}$). This causes the processing time of the $i-1$st processor ($\alpha_{i-1} w_{i-1} T_{cp}$) to be a negative value where the $i-1$st processor finishes processing its load even before it receives the load. Therefore, the above inequality is true. □

There thus exists a crossover point across the two lines and the optimal processing time is achieved at that point where $T_1 = T_2 = \cdots = T_i$ as in Fig. 5.

One can see that this procedure can be continued up to the case including all the finish times, $T_1, T_2, \ldots, T_N$. Then $T_1 = T_2 = \cdots = T_N$ will be obtained to minimize the solution time. Hence the minimal solution time involves all processors stopping their computing at the same time.

Now $T_1 = T_2 = \cdots = T_{i-1}$ and T_i can be rewritten by

$$T_1 = T_2 = \cdots = T_{i-1} = (\alpha_2 + \alpha_3 + \cdots + \alpha_{i-1})ZT_{cm} + \alpha_{i-1} w_{i-1} T_{cp} \quad (38)$$

$$T_i = (\alpha_2 + \alpha_3 + \cdots + \alpha_i)ZT_{cm} + \alpha_i w_i T_{cp} \quad (39)$$

Since $T_1 = T_2 = \cdots = T_i$, α_i can be expressed as a function of $\alpha_{i-1}, \alpha_{i-2}, \ldots$ and α_1.

$$\begin{aligned} \alpha_i &= \frac{w_{i-1}T_{cp}}{ZT_{cm} + w_iT_{cp}}\alpha_{i-1} \\ &= k_{i-1}\alpha_{i-1} \\ &= k_{i-1}k_{i-2}\alpha_{i-2} \\ &\vdots \\ &= k_{i-1}k_{i-2}\cdots k_1 \cdot \alpha_1 \qquad 2 \le i \le N \end{aligned} \tag{40}$$

where

$$k_j = \frac{\alpha_{j+1}}{\alpha_j} = \frac{w_jT_{cp}}{ZT_{cm} + w_{j+1}T_{cp}} \qquad 1 \le j \le N-1$$

Since the sum of α_i's must be one, α_1 can be obtained by the normalization equation.

$$\begin{aligned} 1 &= \alpha_1 + \alpha_2 + \alpha_3 + \cdots + \alpha_N \\ &= (1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{N-1})\alpha_1 \end{aligned} \tag{41}$$

From the above the optimal values of α_i's that the originating processor should calculate in order to achieve the minimum solution time can be computed by the following algorithm (which appears in a slightly different form in [18].)

1) $$k_j = \frac{w_jT_{cp}}{ZT_{cm} + w_{j+1}T_{cp}} \qquad 1 \le j \le N-1 \tag{42}$$

2) $$\begin{aligned} \alpha_1 &= [\,1 + k_1 + k_1k_2 + \cdots + k_1k_2\cdots k_{N-1}\,]^{-1} \\ &= [\,1 + \sum_{i=1}^{N-1}(\prod_{j=1}^{i} k_j)\,]^{-1} \end{aligned} \tag{43}$$

3) $$\begin{aligned} \alpha_i &= k_1k_2\cdots k_{i-1} \cdot \alpha_1 \\ &= (\prod_{j=1}^{i-1} k_j) \cdot \alpha_1 \qquad 2 \le i \le N \end{aligned} \tag{44}$$

Interestingly, the solution for the optimal load allocations is of a *product form*. That is, the solution of α_i (Eq.(44)) can be expressed as a product of system constants (k_i's) and a normalization constant, α_1. The existence of a product form solution for this deterministic problem is all the more interesting as product form solutions are after associated with the stochastic environment of certain classes queueing networks [22].

III Alternative Architectures

It is possible that there may be different types of bus-oriented architectures [10]. One alternative architecture would include *a control processor* that receives a burst of measurement data and distributes it amongst N processors through a bus. The control processor does no processing itself. Another possibility would be the case of a network without control processor and where the processors are not equipped with front-end processors so that the processors cannot communicate and compute simultaneously. The originating processor starts computing after it finishes distributing load to the other processors.

The proof for these cases that to achieve a minimum solution time all processors must finish their processing load at the same time is similar to that in this paper and is the subject of a technical report [15].

The algorithm for computing the optimal values of α_i's is the same as in this paper in both cases except $k_{N-1} = \frac{w_{N-1}}{w_N}$ for a network where processors are not equipped with front-end processors and where the originating processor is processor N.

IV Conclusion

Proofs now exist that the minimal finish time for load sharing a divisible job on a bus network and linear daisy chain network [14] involves having all the processors stop at the same time. An open problem is the demonstration of a similar result for tree networks [9].

References

[1] R.R. Tenney and N.R. Sandell,Jr., "Detection with distributed sensors," *IEEE Transaction on Aerospace and Electronic Systems*, vol. AES-17, pp. 501–510, July 1981.

[2] C.Y. Chong, E. Tse, and S. Mori, "Distributed estimation in networks," presented at the American Control Conference, San Franciso, 1983.

[3] S.H. Bokhari, *Assignment Problems in Parallel and Distributed Computing*, Boston: Kluwer Academic Publishers, 1987.

[4] H.S. Stone, "Multiprocessor scheduling with the aid of network flow algorithms," *IEEE Transaction on Software Engineering*, vol. SE-3, no. 1, pp. 85–93, Jan. 1977.

[5] J. Du and J.Y.T. Leung, "Complexity of scheduling parallel task systems," *SIAM Journal on Discrete Mathematics*, pp. 473–487, Nov. 1989.

[6] J. Blazewicz, M. Drabowski, and J. Weglarz, "Scheduling multiprocessor tasks to minimize schedule length," *IEEE Transactions on Computers*, vol. C-35, pp. 389–398, May 1986.

[7] W. Zhao, K. Ramamritham, and J.A. Stankovic, "Preemptive scheduling under time and resource constraints," *IEEE Transactions on Computers*, vol. C-36, pp. 949–960, Aug. 1987.

[8] Y.C. Cheng and T.G. Robertazzi, "Distributed computation with communication delays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 6, pp. 700–712, Nov. 1988.

[9] Y.C. Cheng and T.G. Robertazzi, "Distributed computation for tree network with communication delays," *IEEE Transactions on Aerospace and Systems*, vol. 26, no. 3, pp. 511–516, May 1990.

[10] S. Bataineh and T.G. Robertazzi, "Distributed computation for a bus networks with communication delays," *Proceedings of the 1991 Conference on Information Sciences and Systems*, The Johns Hopkins University, Baltimore, pp. 709–714, March 1991.

[11] S. Bataineh and T.G. Robertazzi, "Bus oriented load sharing for a network of sensor driven processors," *IEEE Transactions on Systems, Man and Cybernetics*, vol.21, no. 5, Sept. 1991.

[12] S. Bataineh and T.G. Robertazzi, "Closed form solutions for bus and tree networks of processors load sharing a divisible job," *SUNY at Stony Brook College of Engineering and Applied Science Technical Report*, no. 627, May 1992. (Available from T. Robertazzi).

[13] S. Bataineh and T.G. Robertazzi, "Ultimate performance limits for networks of load sharing processors," *Proceedings of the 1992 Conference on Information Sciences and Systems*, Princeton, NJ, pp. 794–799, March 1992.

[14] T.G. Robertazzi, "Processor equivalence for load sharing processor daisy chains," *accepted by the IEEE Transactions on Aerospace and Electronic Systems for Oct. 1993 issue.*

[15] J. Sohn and T.G. Robertazzi, "Optimal load sharing for a divisible job on bus networks," *SUNY at Stony Brook College of Engineering and Applied Science Technical Report.* (Available from T. Robertazzi).

[16] H.J. Kim, G.I. Jee, and J.G. Lee, "Optimal load distribution for tree network processors," submitted for publication.

[17] D. Ghose and V. Mani, "Distributed computation in a linear network: closed form solution and computational techniques," submitted for publication.

[18] V. Bharadwaj, D. Ghose, and V. Mani, "Closed form solutions for optimal processing time in distributed single-level tree networks with communication delays," submitted for publication.

[19] V. Bharadwaj, D. Ghose, and V. Mani, "A new strategy of load distribution in a distributed single-level tree network with communication delays," submitted for publication.

[20] V. Bharadwaj, D. Ghose, and V. Mani, "An efficient load distribution strategy for a distributed linear network of processors with communication delays," submitted for publication.

[21] D. Ghose, and V. Mani, "Distributed computation with communication delays: Asymptotic performance analysis," submitted for publication.

[22] F. Baskett, K.M. Chandy, R.R. Muntz, and F. Palacios, "Open, closed and mixed networks of queues with different classes of customers," *Journal of the ACM*, vol. 22, no. 2, pp. 248–260, April 1975.

Figure Captions

Figure 1. Bus network with N processors.

Figure 2. Timing diagram for bus interconnected processors with front-end processors.

Figure 3. T_1 and T_2 as a function of α_1.

Figure 4. $T_1 = T_2$ and T_3 as a function of α_2.

Figure 5. $T_1 = T_2 = \cdots = T_{i-1}$ and T_i as a function of α_2.

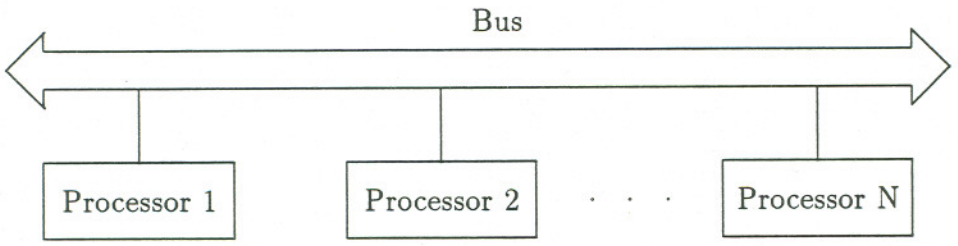

Figure 1. Bus network with N processors.

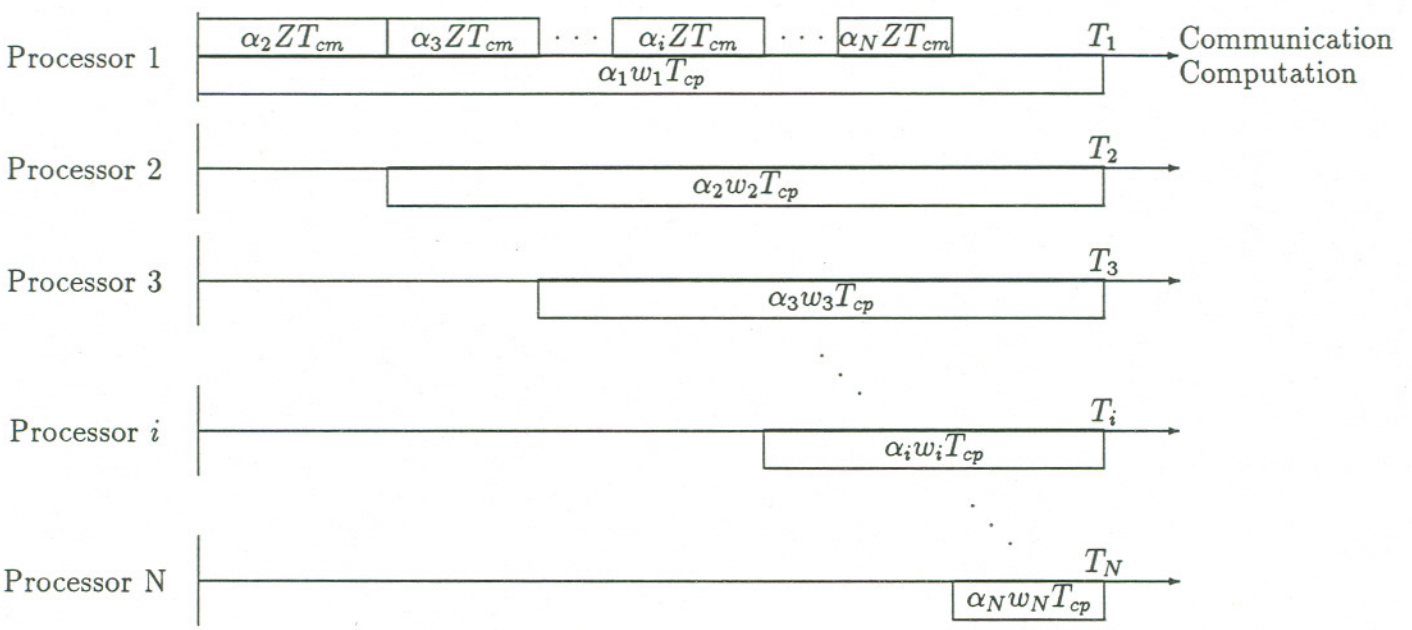

Figure 2. Timing diagram for bus interconnected processors with front-end processors.

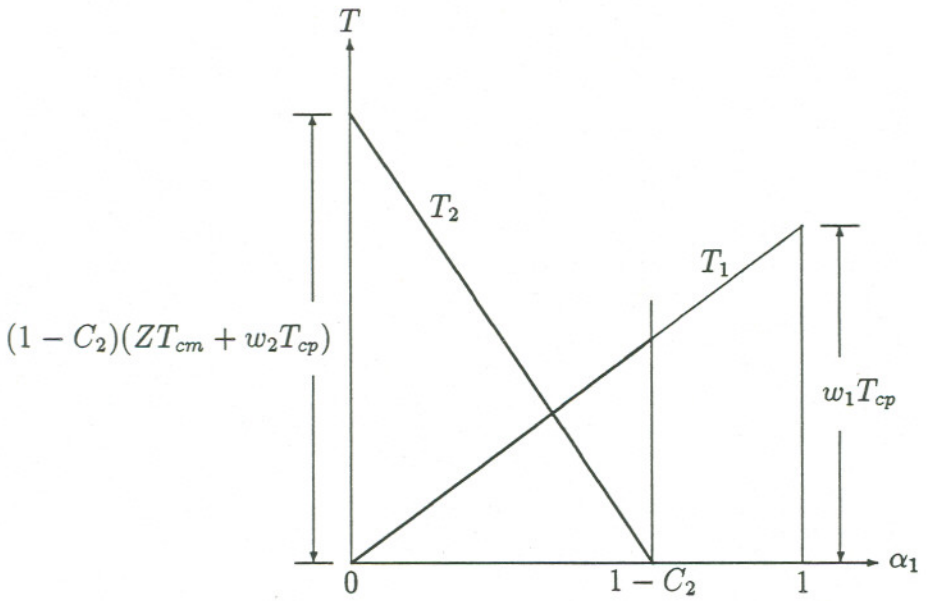

Figure 3. T_1 and T_2 as a function of α_1.

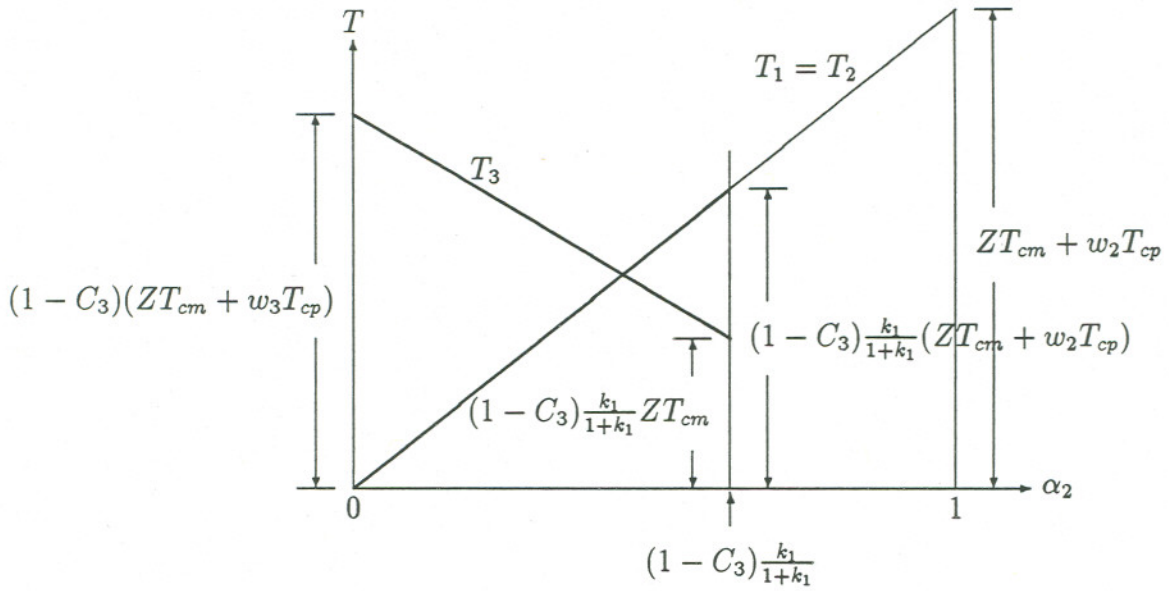

Figure 4. $T_1 = T_2$ and T_3 as a function of α_2.

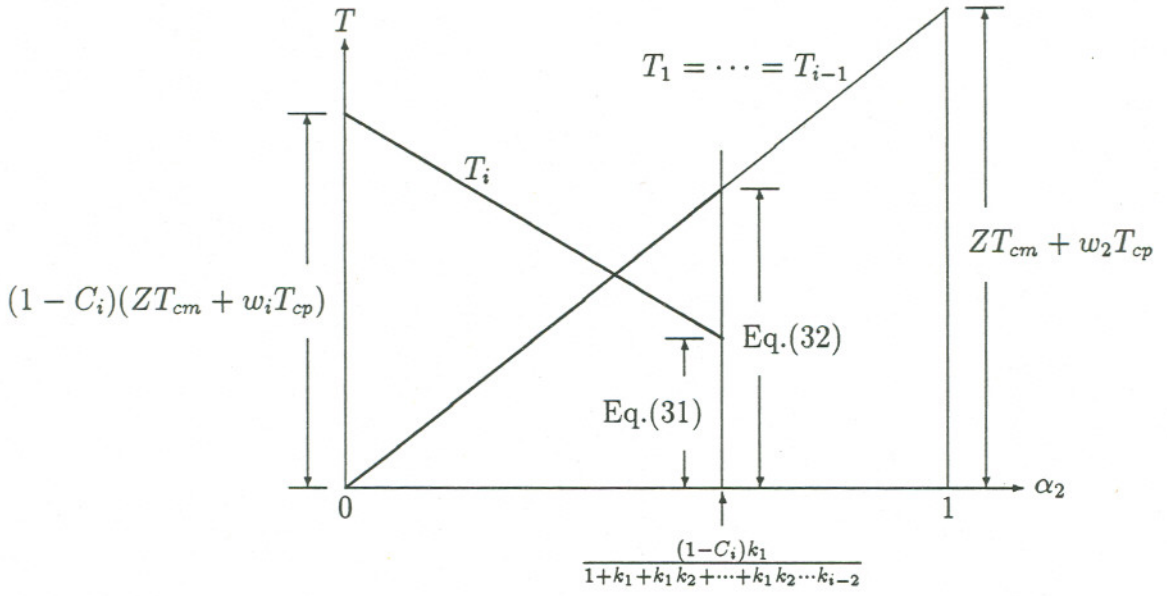

Figure 5. $T_1 = T_2 = \cdots = T_{i-1}$ and T_i as a function of α_2.