통신지연이 있는 분산처리의 최적화

김형중 · 지규인 · 이장규
(강원대학교 제어계측공학과) (서울대학교 제어계측공학과)

Optimal Distributed Computation with Communicaction Delays

Hyoung Joong Kim, Gyu-In Jee
Control & Instrumentation Engineering
Kangweon National University
and
Jang Gyu Lee
Control & Instrumentation Engineering
Seoul National University

Abstract

Tree network consisting of communicating processors is considered. The objective is to minimize the computation time by distributing the processing load to other nodes. The effect of the order of load distribution on the processing time is addressed. An algorithm which optimally determines the order of load distribution is developed. It is shown that the order depends only on the channel capacity between nodes but not on the computing capability of each node.

1 INTRODUCTION

Suppose that a processor, the root node, of the tree network of communicating processors receives a burst of processing load. In order to process the load in a minimal amount of time, all the nodes of the tree network share the processing load given to the root node for utilizing the distributed computation. The processing load is distributed to each child node from its parent node.

The problem of optimal distribution of the processing load among the nodes in the tree network was discussed by Cheng and Robertazzi[1]. They proposed a bottom-up algorithm for an optimal distribution of processing load, in the sense that it minimizes the total processing time. Their algorithm is based on the fact that in order to obtain maximum parallelism and a minimum time solution all processors must stop computing at the same time. But they failed to recognize that different load distribution order can change the total computing time. Actually their algorithm results in the optimal load distribution when the order of load distribution is determined *a priori*.

In this paper we extend their results by considering the load distribution sequence and propose an optimal load distribution algorithm.

2 Problem Statement

Consider a tree network consisting of communicating processors as shown in Figure 1. It is assumed that every node in the tree network

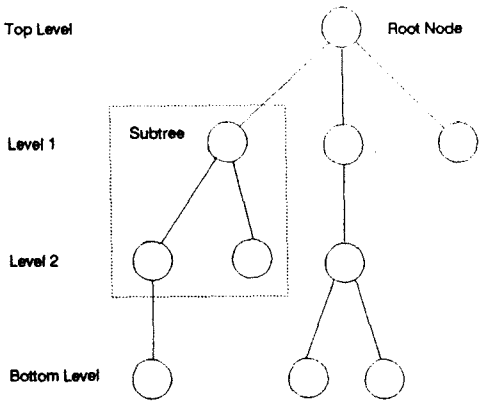

Figure 1: Example of Tree Network

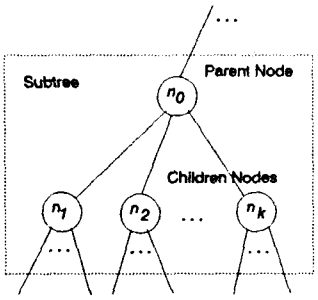

Figure 2: Subtree of Tree Network

can communicate with only its parent node and children nodes and a node cannot communicate with more than one node at the same time. In general, each node in the network has different computing capability and the channel capacity between every two nodes is different.

Suppose a huge amount of processing load is given to one processor, the root node. In order to process the load effectively, so minimize the total processing time, the load is distributed over the whole nodes for utilizing the distributed computing. The root node in top level first keep some fraction of the total processing load and distributes the remaining load to its children nodes in the next lower level. The nodes in the lower level keep some fraction of what they have received and distribute the remaining load to their children nodes in the next lower level, and so on. This distribution proceeds until the nodes in the bottom level are reached.

Consider a subtree of the tree network consisting of one parent node, n_0, and k children nodes, $n_1, n_2, \ldots, n_k$ as shown in Figure 2. The parent node, n_0, which has received some load L from its parent node keeps α_0 fraction of L for itself to process and distributes the remaining load to its k children nodes one by one in a specific order. Define d_i as the ith distributed child node. For example, if parent node distributes some amount of L to the child node n_j in the ith place then $d_i = n_j$. Note that $d_0 = n_0$. Suppose the ith distributed child node d_i receives a fraction α_i of L. On receiving the α_i fraction of L, the ith distributed child node d_i keeps β_i fraction of what it has just received and distribute the remaining load to its children nodes. Note that index i of α_is and β_is correspond to not the position of node as in [1] but the order of distribution. In [1] α_i is the fraction of L which node n_i receives and β_i is the fraction of what node n_i has received which node n_i keeps.

With these differently defined coefficients Cheng and Robertazzi's load distribution algorithm [1] for the tree network can be given by following equations:

$$\alpha_0 w_0 T_{cp} = \alpha_1 z_1 T_{cm} + \alpha_1 w_1 \beta_1 T_{cp} \quad (1)$$

$$\alpha_i w_i \beta_i T_{cp} = \alpha_{i+1} z_{i+1} T_{cm} + \alpha_{i+1} w_{i+1} \beta_{i+1} T_{cp}, \quad \text{for} \quad i = 1, 2, \ldots, k-1 \quad (2)$$

$$\alpha_0 + \alpha_1 + \cdots + \alpha_k = 1 \quad (3)$$

Here w_i is a coefficient inversely proportional to the computation speed of the ith distributed child node d_i, z_i is a coefficient inversely proportional to the channel speed between the parent node and the child node d_i, T_{cp} is the time it takes for one node to process the entire processing load when the corresponding w is equal to 1, and T_{cm} denotes the time to transmits the entire

processing load over the channel when the corresponding z is equal to 1. Since it is a bottom-up algorithm, values of β_is have been previously obtained from the next lower level using (1)-(3) where β_i corresponds to α_0 of the subtree which has node d_i as its parent node.

The equations (1)-(3) can be rewritten by following equations:

$$\alpha_i = \gamma_i \alpha_0, \quad \text{for} \quad i = 1, 2, \ldots, k \quad (4)$$

$$\sum_{i=0}^{k} \alpha_i = 1 \quad (5)$$

where

$$\gamma_i \triangleq \frac{b_0}{a_1 + b_1} \frac{b_1}{a_2 + b_2} \cdots \frac{b_{i-1}}{a_i + b_i} w_0 \quad (6)$$

$$a_i \triangleq z_i \frac{T_{cm}}{T_{cp}} \quad (7)$$

$$b_i \triangleq w_i \beta_i, \quad \text{for} \quad i = 1, 2, \ldots, k \quad (8)$$

$$b_0 \triangleq 1 \quad (9)$$

For given values of T_{cm}, T_{cp}, z_is, w_is, and β_is, the optimal load distribution, *i.e.* α_is can be determined by the equations (4)-(5). Once α_is are obtained, the total processing time, which is just the processing time of the root node is given by $\alpha_0 w_0 T_{cp}$. Note that it can be considered as a function of α_0 only, for given w_0 and T_{cp}. Now, what if we change the order of distribution? Can we expect to have a different, hopefully smaller, value of α_0? The next example will answer this question.

Example 1: Consider a subtree network shown in Figure 3. The corresponding values of a_is and b_is for each child node are given. First, assume that the parent node n_0 distributes its processing load to its children nodes in turn from left to right, that is, $d_1 = n_1$, $d_2 = n_2$, $d_3 = n_3$, and $d_4 = n_4$. This is the distribution order Cheng and Robertazzi considered. In this case, $a_1 = 20$, $a_2 = 10$, $a_3 = 5$, $a_4 = 1$, and $b_1 = b_2 = b_3 = b_4 = 1$. From (4)-(5), $\alpha_5 = 0.9496$. Next, consider another distribution order, from right to left. Then, $d_1 = n_4$, $d_2 = n_3$, $d_3 = n_2$, and $d_4 = n_1$. In this case, $a_1 = 1$, $a_2 = 5$, $a_3 = 10$, $a_4 = 20$, and $b_1 = b_2 = b_3 = b_4 = 1$. This results in a smaller $\alpha_0 = 0.6284$ than the previous one. So we can process the total processing load faster.

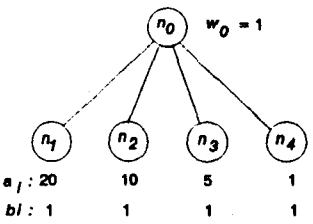

Figure 3: Subtree with 4 Children Nodes

From the above example, it can be observed that the order of load distribution over the children nodes is an important issue when we consider the distributed computing for the tree network. If a parent node has k children nodes, there are $k!$ possible orders of load distribution. In the next section an optimal load distribution algorithm obtaining the fastest processing speed is presented.

3 Optimal Distribution of Processing Load

The main objective of this paper is to achieve the minimum total processing time by determining the optimal order of load distribution from the parent node to children nodes. If a parent node n_0 has k children nodes, there are $k!$ possible sequences of load distribution. Since the total processing time is just the processing time of the root node $\alpha_0 w_0 T_{cp}$, and w_0 and T_{cp} are known values for each given subtree, the objective is equivalent to find the optimal distribution order resulting in the minimum α_0 among the $k!$ possible orders.

As shown in Example 1 we can consider α_0 as a function of distribution sequence $I = \{I_1, I_2, \ldots, I_k\}$, $\alpha_0(I)$ for a given subtree. If the

parent node of subtree in Figure 2 distribute the processing load from left to right, that is, $d_1 = n_1$, $d_2 = n_2$, $d_3 = n_3$, $d_4 = n_4$ then $I = \{1,2,3,4\}$. Define an operator $S_{i,j}$ on sequence I as follows:

$$S_{i,j}(I) \triangleq \{I_1, \ldots, I_{i-1}, I_j, I_{i+1}, \ldots, I_{j-1}, I_i, I_{j+1}, \ldots, I_k\} \quad (10)$$

That is, $S_{i,j}$ swaps the ith and the jth entries of I.

Lemma 1 *For any subtree consisting of one parent node, n_0, and k children nodes, n_1, n_2, ..., n_k and a given sequence of load distribution order I, if $a_i \geq a_{i+1}$ then*

$$\alpha_0(I) \geq \alpha_0(S_{i,i+1}(I)), \quad \text{for} \quad i = 1, \ldots, k-1 \quad (11)$$

The equality is satisfied when $a_i = a_{i+1}$.

Proof: From equations (4)-(5), $\alpha_0(I)$ can be given by

$$\alpha_0(I) = \frac{1}{1+q} \quad (12)$$

where

$$q \triangleq \gamma_1 + \gamma_2 + \ldots + \gamma_k \quad (13)$$

$$= \frac{b_0}{a_1+b_1}w_0 + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}w_0 + \ldots + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}\cdots\frac{b_{i-1}}{a_i+b_i}w_0 + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}\cdots\frac{b_{i-1}}{a_i+b_i}\frac{b_i}{a_{i+1}b_{i+1}}w_0 + \ldots + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}\cdots\frac{b_{k-1}}{a_k+b_k}w_0 \quad (14)$$

Let $I' = S_{i,i+1}(I)$. Then $\alpha_0(I')$ can be written by

$$\alpha_0(I') = \frac{1}{1+q'} \quad (15)$$

where

$$q' = \frac{b_0}{a_1+b_1}w_0 + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}w_0 + \ldots + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}\cdots\frac{b_{i-1}}{a_{i+1}+b_{i+1}}w_0 + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}\cdots\frac{b_{i-1}}{a_{i+1}+b_{i+1}}\frac{b_{i+1}}{a_i+b_i}w_0 + \ldots + \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}\cdots\frac{b_{k-1}}{a_k+b_k}w_0 \quad (16)$$

From equations (14) and (16)

$$\alpha_0(I) - \alpha_0(I') = \frac{1}{(1+q)(1+q')}(q'-q) \quad (17)$$

and

$$q' - q = \frac{b_0}{a_1+b_1}\frac{b_1}{a_2+b_2}\cdots\frac{b_{i-1}}{a_i+b_i} \cdot \frac{1}{a_{i+1}+b_{i+1}}(a_i - a_{i+1}) \quad (18)$$

Therefore $\alpha_0(I) \geq \alpha_0(I')$ when $a_i \geq a_{i+1}$ and $\alpha_0(I) = \alpha_0(I')$ when $a_i = a_{i+1}$. ■

Now, we can establish an optimal load distribution algorithm achieving the fastest processing speed. It is stated as follows:

Load Distribution Algorithm

Step 1. Determine the distribution order such that the child node having smaller a_i, i.e. faster channel speed receives the fraction of processing load first all the way through.

Step 2. Calculate the fraction of processing load α_i for each child node d_i by using equations (4)-(8).

Note that once the distribution is determined according to Load Distribution Algorithm, $a_1 \leq a_2 \leq \ldots \leq a_k$. Between a_is and b_is of children nodes only a_i values, especially the relative magnitudes of a_is, determine the optimal order of load distribution. That is, we don't need to consider the computation speed of the child node to determine the load distribution sequence.

Theorem 1 *Load Distribution Algorithm provides the optimal load distribution order in the sense of minimizing the total processing time.*

Proof: We can prove this by contradiction. Without loss of generality we can assume that all a_is are different each other. Suppose that any distribution order except the one with ascending order of magnitude of a_is results in the smallest α_0 among all the possible $k!$ distribution sequences. Since this distribution sequence is not distributed in the ascending order all the

way through there always exists at least one pair of neighboring node where $a_i > a_{i+1}$. Therefore from Lemma 1 we can have larger value of α_0 by swapping that neighboring nodes d_i and d_{i+1}. This contradicts the original assumption of an optimal load distribution sequence. ■

4 Conclusion

Load distribution problem for tree network of communicating processors is considered. The processing load is distributed to the other nodes to utilize the distributed computing. Cheng and Robertazzi's algorithm for optimal load distribution to minimize the total processing time is generalized by considering the order of load distribution sequence. The effect of the order of distribution on the computation time is discussed and the optimal load distribution order minimizing the total processing time is derived. It is shown that the order depends only on the channel capacity between nodes, but not on the computing capabilities of each node.

References

[1] Cheng, Y. C., and Robertazzi, T. G. (1990) Distributed Computation for a Tree Network with Communication Delays
IEEE Transactions on Aerospace and Electronics Systems,
AES-26, 3 (May 1990), pp. 511-516.