Divisible Load Cut Through Switching in Sequential Tree Networks

JUI TSUN HUNG

THOMAS G. ROBERTAZZI, Senior Member, IEEE
SUNY Stony Brook

Recursive analytical expressions for speedup and solution time for a multilevel tree sequentially processing a divisible load under cut through switching are developed. Such cut through switching is shown to be more efficient than store and forward switching. Aerospace applications include sensor networks, radar, and satellite imagery processing.

Manuscript received August 6, 2003; revised December 9, 2003 and March 23, 2004; released for publication March 24, 2004.

IEEE Log No. T-AES/40/3/835892.

Refereeing of this contribution was handled by M. G. Simoes.

This work was supported by the NSF under Grant CCR-99-12331.

Authors' current addresses: J. T. Hung, Memes Technology Corp., Nangkang Software Park, Taipei, Taiwan; T. G. Robertazzi, Cosine Laboratory, Dept. of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, E-mail: (tom@ece.sunysb.edu).

0018-9251/04/$17.00 © 2004 IEEE

I. INTRODUCTION

The processing of massive amounts of data on distributed and parallel networks is becoming more and more common. Aerospace applications include radar and infrared tracking, satellite imaging, and sensor networks. Over the past 15 years, a number of researchers have mathematically modeled such processing using a divisible load scheduling model [1–3] that is useful for data parallelism applications.

Divisible loads are ones that consist of data that can be arbitrarily partitioned among a number of processors interconnected through some network. Divisible load modeling usually assumes no precedence relations among the data. For instance, a satellite image processing system may involve a stream of independent images arriving at a robotic tape silo for storage and then being distributed to processors on a cluster computer. In this situation divisible load scheduling theory can answer the question of how to optimally schedule image transmissions and assign load to processors to maximize parallel system speedup and minimize overall processing time.

One reason that motivated the creation of divisible load theory was the need for a means of modeling integrated measurements, communication, and computation in sensor networks [4]. At the time of [4] there was a recognized need to combine the communication and computation aspects of sensor networks modeling. Divisible load theory allows such integration. Because of its underlying linear model, continuous mathematics framework, divisible load modeling is very tractable. It has been used to accurately and directly model such features as specific network topologies and scheduling policies [2, 4–11, 14] computation versus communication load intensity [2, 4], time-varying inputs [15], multiple job submission [2, 16–18], and numerous applications such as image processing, databases, and multimedia.

A. Our Contribution

The goal of this paper is to quantify the superiority of virtual cut through switching to store and forward switching for networks processing divisible loads. Under store and forward like switching, a node in a tree must receive load for all of its descendants before beginning to distribute load to them [19]. This clearly can be improved upon. The way to do this is to use virtual cut through switching [12, 13] which allows a node to distribute load to its descendants even as it continues to receive its descendants' load. Both strategies are used here, which ends with a comparison. Note that the majority of the divisible load scheduling literature to date involves store and forward switching.

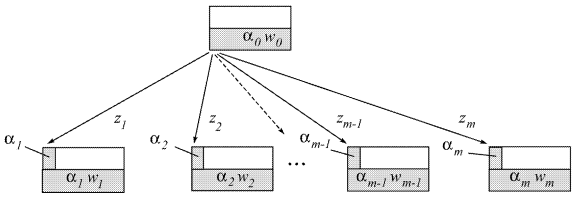


Fig. 1. Structure of a heterogeneous single level tree with sequential distribution, simultaneous start.

To achieve our goal we discuss tree networks. Tree networks, though of interest in their own right, can be used to span any interconnection topology to deliver load to processors. They are thus relevant for many load distribution policies in such commonly used interconnection networks as hypercubes and meshes. Moreover a logical (Ethernet or wireless channel like) bus can be modeled as a single level tree (root and adjacent children only) with load distributed sequentially to children and all links having the same transmission speed. Multilevel trees can model cascaded networks as discussed in Section IIIB.

A multilevel tree is considered here where at time $t = 0$ the divisible load is available at the root node. Load is optimally (in a solution time or speedup sense) distributed throughout the tree to gain the benefit of parallel processing of the load. Our main results are closed-form solutions under a variety of scheduling scenarios (including the use of fat trees) for the solution time and speedup of single level tree networks and recursive expressions for homogeneous multilevel tree networks. These recursive expressions are novel. Calculating speedup for heterogeneous trees is also discussed.

This paper begins with the development of some notation and analytic background in Section II. The scheduling method using cut through switching from level to level is derived in Section III. The scheduling method using store and forward switching from level to level is derived in Section IV. In Section V we compare the performance of cut through switching scheduling and store and forward scheduling. The conclusion appears in Section VI.

II. MODEL AND NOTATION

A. Model and Notation for Single Level Tree

If a node begins to process its load as soon as the load begins to be received, we call this simultaneous start. It is illustrated in Fig. 1 (where each node contains a miniature timing diagram—such diagrams are explained later.)

For a heterogeneous single level tree, which can be collapsed into an equivalent node, the notation is presented as follows.

α_0 Load fraction assigned to root processor
α_i Load fraction assigned to ith link processor pair
w_i Inverse computing speed on ith processor
w_{eq} Inverse computing speed of equivalent node collapsed from single level tree
z_i Inverse link speed on ith link
T_{cp} Computing intensity constant. Entire load can be processed in $w_i T_{cp}$ seconds on ith processor
T_{cm} Communication intensity constant. Entire load can be transmitted in $z_i T_{cm}$ seconds over ith link
$T_{f,m}$ Finish time of an equivalent node collapsed from single level tree composed of one root node and m children nodes. $T_{f,m}$ is equal to $w_{eq} T_{cp}$
$T_{f,0}$ Finish time for entire divisible load solved on root processor, (i.e., tree without any children nodes but the root node). $T_{f,0}$ is equal to $1 \times w_0 T_{cp}$, that is $w_0 T_{cp}$.

DEFINITION 1 First γ_{eq} is the ratio of the inverse computing speed on an equivalent node to that on the root node

$$\gamma_{eq} = w_{eq}/w_0 = T_{f,m}/T_{f,0}. \tag{1}$$

An equivalent node can exactly present the operation characteristics of a subnetwork it replaces. Equivalent element modeling is a feature of linear models, such as circuit theory, queueing theory, and divisible load theory.

DEFINITION 2 Speedup, the ratio of finish time on one processor (i.e., the root node) to that on an equivalent node collapsed from a single level tree. It is thus a measure of parallel processing advantage. This value is equal to the ratio of the inverse computing speed on the root node to that on an equivalent node, i.e., the inverse of γ_{eq}. Hence,

$$\text{Speedup} = T_{f,0}/T_{f,m} = w_0/w_{eq} = 1/\gamma_{eq}. \tag{2}$$

B. Model and Notation for Multilevel Tree

A heterogeneous multilevel tree network is too complex to obtain a closed-form solution of speedup. Therefore, a homogeneous multilevel tree network where root processors are equipped with a front-end processor for off-loading communications is evaluated.

Suppose that after a subroot receives all of the assigned fraction of load for its descendants, it starts distributing these loads to its descendants concurrently. This strategy is called "store and forward switching with simultaneous distribution" (see Fig. 2).

The notation for a multilevel homogeneous fat tree is denoted as follows.

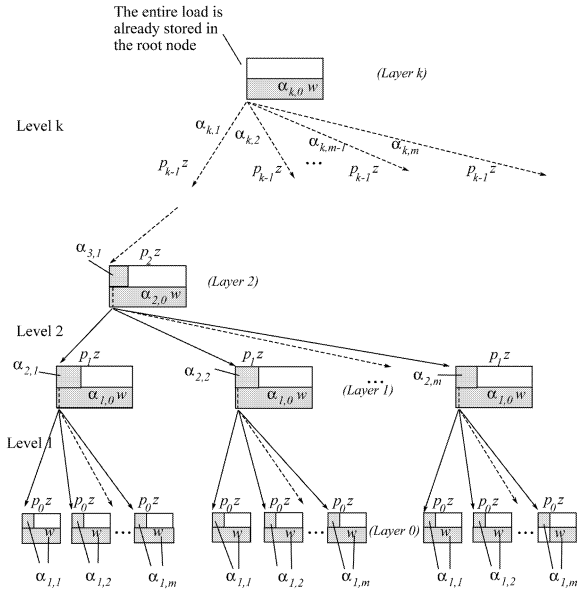


Fig. 2. Structure of multilevel homogeneous tree using cut through switching, sequential distribution, simultaneous start.

$\alpha_{j,0}$ Load fraction assigned to root processor of jth level subtrees

$\alpha_{j,i}$ Load fraction assigned to ith link-processor pair of jth level subtrees

$w_{i_{eq_{j-1}}}$ Inverse computing speed of equivalent ith node which represents $(j-1)$th level subtree, which consists of collapsed single level subtrees from level 1 ascending to level $j-1$. In a homogeneous multilevel tree, we assume that $w_{eq_{j-1}} = w_{i_{eq_{j-1}}}$ $(i = 1,2,\ldots,m)$

$T_{f,m}^{h,k}$ Processing finish time of k level homogeneous tree with one root node and m equivalent children nodes.

DEFINITION 3 $p_{j-1,i}$; the multiplier of the inverse capacity of the ith link at level j (see Fig. 2). The value of the multiplier $p_{j-1,i}$ is defined as the inverse of the total number of children processor descendants at and below layer $j-1$ for the ith subtree. The variable $p_{j-1,i}$ allows fat tree modeling. A fat tree allocates more capacity to nodes near the root to improve the transmission speed. In a homogeneous multilevel fat tree, $p_{j-1} = p_{j-1,i}$ $(i = 1,2,\ldots,m)$. Hence,

$$p_{j-1} = \left(\sum_{l=0}^{j-1} m^l\right)^{-1}, \qquad 0 < p_{j-1} \le 1. \tag{3}$$

With this choice of p_{j-1}, the transmission capacity between a level j parent node and its children nodes is $1/(p_{j-1}z)$, which is larger than the capacity of bottommost level links by $1/p_{j-1}$. This implicitly indicates that each node within an equivalent subtree from layer $j-1$ down to layer 0 has an equivalent transmission capacity of $1/z$ to the root.

DEFINITION 4 γ_j, the ratio of the inverse computing speed on an equivalent node at level j to that on the root node

$$\gamma_j = w_{eq_j}/w. \tag{4}$$

DEFINITION 5 Speedup, the ratio of finish time on one processor (i.e., the root node) to that on an equivalent node collapsed from a subtree from level k to level 1. This value is also equal to the ratio of the inverse computing speed on the root node to that on an equivalent node, i.e., the inverse of γ_k. Hence,

$$\text{Speedup} = T_{f,0}/T_{f,m}^{h,k} = w/w_{eq_k} = 1/\gamma_k. \tag{5}$$

III. CUT THROUGH SWITCHING WITH SEQUENTIAL DISTRIBUTION

For the purposes of determining the optimal load allocations, the single level trees within the overall multilevel tree are divided into a root single level subtree (level k) and single level subtrees below the root subtree (level $1,2,\ldots,k-1$). It is assumed that all data is available (and stored) at the root at $t = 0$. Thus the root can immediately deliver load to its children at level k. This is the root node with data storage case.

For the other single level trees, it is assumed that a root's load must be completely received by a single level tree root before load is distributed to its children. After this, load is relayed through the root to its children in virtual cut through mode. This is the root node without data storage case. Note that the root node without data storage policy is particularly appropriate when bandwidth is limited, as in a homogeneous tree.

In both cases, only the simultaneous start strategy is considered. In the simultaneous start strategy each processor begins processing the received data while it continues to receive the data. This strategy was originally described by Kim [20].

In the following both cases are examined, first in the context of a single level tree in isolation and then in the context of multilevel trees.

In this section we evaluate the performance of the multilevel tree model using sequential distribution model under the cut through switching and the simultaneous start strategy. The complexity of sequential distribution model is more involved than that of the better performing simultaneous distribution. This model is applied to situations when the parent nodes do not have enough capacity and ability to enable the simultaneous distribution model.

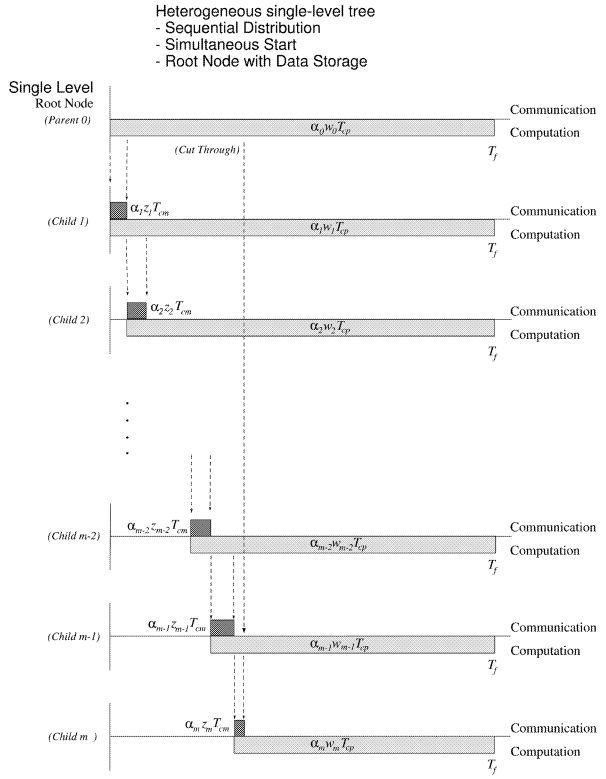


Fig. 3. Timing diagram of single level tree with sequential distribution, simultaneous start, and root node with data storage.

A. Processors Using Simultaneous Start Model in a Single Level Tree

The following two subsections discuss sequential distribution in single level trees: one model with data storage and the second model without data storage.

1) *Single Level Tree: Root Node with Data Storage*: The process of sequential load distribution can be represented by Gantt chart-like timing diagrams, as illustrated in Fig. 3. Here the horizontal axis is time and communication appears above the axis and computation appears below the axis. According to the figure, the fundamental recursive equations of the system can be formulated as follows:

$$\alpha_0 w_0 T_{cp} = \alpha_1 w_1 T_{cp} \tag{6}$$

$$\alpha_{i-1} w_{i-1} T_{cp} = \alpha_{i-1} z_{i-1} T_{cm} + \alpha_i w_i T_{cp}, \qquad i = 2,3,\ldots,m. \tag{7}$$

The normalization equation for the single level tree with intelligent root is

$$\alpha_0 + \alpha_1 + \alpha_2 + \cdots + \alpha_m = 1. \tag{8}$$

This gives $m+1$ linear equations with $m+1$ unknowns. From (6)

$$\alpha_0 = \frac{w_1}{w_0}\alpha_1 = \frac{1}{k_1}\alpha_1 \qquad \text{where} \quad k_1 = w_0/w_1. \tag{9}$$

From (7)

$$\alpha_i = \frac{w_{i-1}T_{cp} - z_{i-1}T_{cm}}{w_i T_{cp}}\alpha_{i-1} = q_i \alpha_{i-1} = \left(\prod_{l=2}^{i} q_l\right) \times \alpha_1, \qquad i = 2,3,\ldots,m \tag{10}$$

where $q_i = (w_{i-1}T_{cp} - z_{i-1}T_{cm})/w_i T_{cp}$. We assume $w_{i-1}T_{cp} > z_{i-1}T_{cm}$; that is, communication time must be faster than computation time. See also [2] for a discussion of the best choices of w_i and z_i. According to (9) and (10), the normalization equation (8) leads to

$$\left[\frac{1}{k_1} + 1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]\alpha_1 = 1 \tag{11}$$

Then the value of α_1 is

$$\alpha_1 = \frac{1}{\frac{1}{k_1} + 1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)} \tag{12}$$

Therefore, the finish solution time is derived as follows:

$$T_{f,m} = \alpha_0 w_0 T_{cp} = \frac{1}{k_1}\alpha_1 w_0 T_{cp} = \frac{1}{1 + k_1\left[1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]} w_0 T_{cp}. \tag{13}$$

Since a single level tree can be collapsed into a single equivalent node, the equivalent inverse computation speed w_{eq} of a collapsed node can be derived as follows:

$$w_{eq}T_{cp} = T_{f,m} = \frac{1}{1 + k_1\left[1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]} w_0 T_{cp}. \tag{14}$$

According to Definition 1 in Section II, γ_{eq} is equal to w_{eq}/w_0. Thus, from (14)

$$\gamma_{eq} = \frac{w_{eq}}{w_0} = \frac{1}{1 + k_1\left[1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]}. \tag{15}$$

Since

$$T_{f,0} = \alpha_0 w_0 T_{cp} = 1 \cdot w_0 T_{cp} \tag{16}$$

where $\alpha_0 = 1$, and speedup is the ratio of computation time on one processor to computation time on the entire tree with m children, we obtain

$$\text{Speedup} = \frac{T_{f,0}}{T_{f,m}} = \frac{1}{\gamma_{eq}} = 1 + k_1\left[1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]. \tag{17}$$

As a special case, consider the situation of a homogeneous network where all children processors have the same inverse computing speed and all links

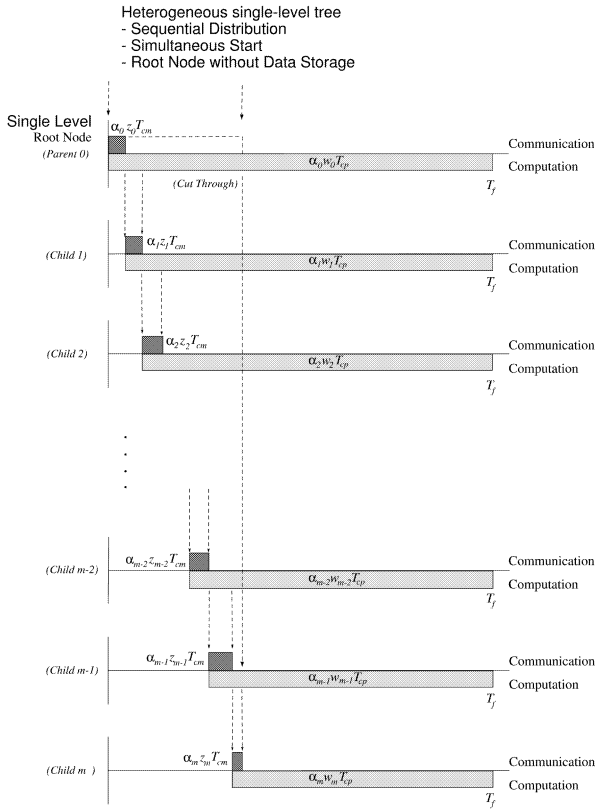


Fig. 4. Timing diagram of single level tree with sequential distribution, simultaneous start, and root node without data storage.

have the same inverse transmission speed (i.e., $w_i = w$ and $z_i = z$ for $i = 1,2,\ldots,m$). Note the root w_0 can be different from w_i. Therefore

$$q_i = \frac{w_{i-1}T_{cp} - z_{i-1}T_{cm}}{w_i T_{cp}} = \frac{wT_{cp} - zT_{cm}}{wT_{cp}} = 1 - \sigma \tag{18}$$

where $\sigma = zT_{cm}/wT_{cp}$ and $i = 2,3,\ldots,m$. Consequently,

$$\text{Speedup} = 1 + \frac{w_0}{w}\left[\frac{1-(1-\sigma)^m}{\sigma}\right]. \tag{19}$$

2) *Single Level Tree: Root Node without Data Storage*: The process of load distribution can be represented by Gantt chart-like timing diagrams, as illustrated in Fig. 4.

The fundamental recursive equations of the system can be formulated as follows:

$$\alpha_0 w_0 T_{cp} = \alpha_1 w_1 T_{cp} + \alpha_0 z_0 T_{cm} \tag{20}$$

$$\alpha_{i-1} w_{i-1} T_{cp} = \alpha_{i-1} z_{i-1} T_{cm} + \alpha_i w_i T_{cp}, \qquad i = 2,3,\ldots,m. \tag{21}$$

The normalization equation for the single level tree with intelligent root is

$$\alpha_0 + \alpha_1 + \alpha_2 + \cdots + \alpha_m = 1. \tag{22}$$

This gives $m + 1$ linear equations with $m + 1$ unknowns. Now from (20)

$$\alpha_0 = \frac{w_1 T_{cp}}{w_0 T_{cp} - z_0 T_{cm}}\alpha_1 = \frac{1}{k_1}\alpha_1 \tag{23}$$

where $k_1 = (w_0 T_{cp} - z_0 T_{cm})/w_1 T_{cp}$.

It is assumed that $w_0 T_{cp} > z_0 T_{cm}$ (communication speed is faster than computation speed for the 0th link and root). Following the similar derivation in Section IIIA1, the expressions of γ_{eq} and speedup can be obtained as follows:

$$\gamma_{eq} = \frac{w_{eq}}{w_0} = \frac{1}{1 + k_1\left[1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]}. \tag{24}$$

Therefore,

$$\text{Speedup} = \frac{T_{f,0}}{T_{f,m}} = \frac{1}{\gamma_{eq}} = 1 + k_1\left[1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]. \tag{25}$$

As a special case, consider the situation of a homogeneous network where all children processors have the same inverse computing speed and all links have the same inverse transmission speed (i.e., $w_i = w$ and $z_i = z$ for $i = 1,2,\ldots,m$). Note the root w_0 can be different from w_i. Then

$$\text{Speedup} = 1 + \frac{w_0 T_{cp} - z_0 T_{cm}}{wT_{cp}}\left[\frac{1-(1-\sigma)^m}{\sigma}\right]. \tag{26}$$

B. Processors with Simultaneous Start. Homogeneous Multilevel Tree Analysis

For purposes of illustration we consider two types of multilevel tree scheduling for homogeneous trees. Here homogeneous trees have processors with identical computing speeds and links with identical transmission speeds. Homogeneous networks arise in practice in such instances as cluster computer installations. Calculating speedup for heterogeneous trees is discussed in Section IIIC.

One type of multilevel tree scheduling discussed here uses sequential distribution of load from a node to its children for all tree levels. This is a good model for a cascaded series of Ethernets (utilizing collision domain interface cards) or a cascaded series of wireless channels. The second type of multilevel tree scheduling uses simultaneous distribution of load from the tree root to its children (in the topmost level of the tree) and sequential distribution for tree levels below that (levels j, $j = 1,2,\ldots,k-1$). By way of example, this can model a large capacity robotic tape silo feeding a number of clusters simultaneously using ATM links. Each cluster consists of one of more cascaded Ethernet implementing sequential

distribution (because of the use of collision domain interface cards).

In this section we develop recursive expressions for solution time and for speedup for the two scenarios. This is done first for levels $j = 1,2,\ldots, k-1$ (with sequential distribution) and then in two subsections for the topmost level k (for both sequential and simultaneous distribution). At the end of these two subsections the overall recursive expressions are presented.

The methodology developed in this section can be applied to other combinations or exclusive uses of scheduling policies at each tree level. The ones discussed here are natural for a first study. Note also that if one is interested solely in the optimal allocations of load to processors, rather than speedup calculations, the methodology of [21] can be used.

1) *Level j Subtree: Root Node without Data Storage*: Consider now a homogeneous multilevel fat tree network where all processors have the same inverse computing speed w and links of level $j+1$ have the inverse transmission speed $p_j z$. The value of p_{j-1} is also defined in Definition 3 in Section II, that is,

$$p_j z = \left[\left(\sum_{l=0}^{j} m^l \right)^{-1} \right] z. \tag{27}$$

The process of load distribution for the multilevel fat tree network using cut through switching for computing and communicating can be represented by Gantt chart-like timing diagrams. According to an equivalent single level fat tree, level j (see Fig. 5), the fundamental recursive equation can formulated as follows:

$$\alpha_{j,0} w T_{cp} = \alpha_{j,1} w_{eq_{j-1}} T_{cp} + \alpha_{j,0} p_j z T_{cm} \tag{28}$$

$$\alpha_{j,i-1} w_{eq_{j-1}} T_{cp} = \alpha_{j,i} w_{eq_{j-1}} T_{cp} + \alpha_{j,i-1} p_{j-1} z T_{cm}$$

$$\text{where} \quad i = 2,3,\ldots,m. \tag{29}$$

The normalization equation for the single level tree with an intelligent root (that can process load as well as distribute it) is

$$\alpha_{j,0} + \alpha_{j,1} + \alpha_{j,2} + \cdots + \alpha_{j,m} = 1. \tag{30}$$

This gives $m+1$ linear equations with $m+1$ unknowns.

Now from (28)

$$\alpha_{j,0} = \frac{w_{eq_{j-1}} T_{cp}}{w T_{cp} - p_j z T_{cm}} \alpha_{j,1} = \frac{1}{k_{eq_{j-1}}} \alpha_{j,1}. \tag{31}$$

Here the expression of $k_{eq_{j-1}}$ is manipulated as follows:

$$k_{eq_{j-1}} = \frac{w T_{cp} - p_j z T_{cm}}{w_{eq_{j-1}} T_{cp}} = \frac{w}{w_{eq_{j-1}}} - \frac{w p_j z T_{cm}}{w_{eq_{j-1}} w T_{cp}} \tag{32}$$

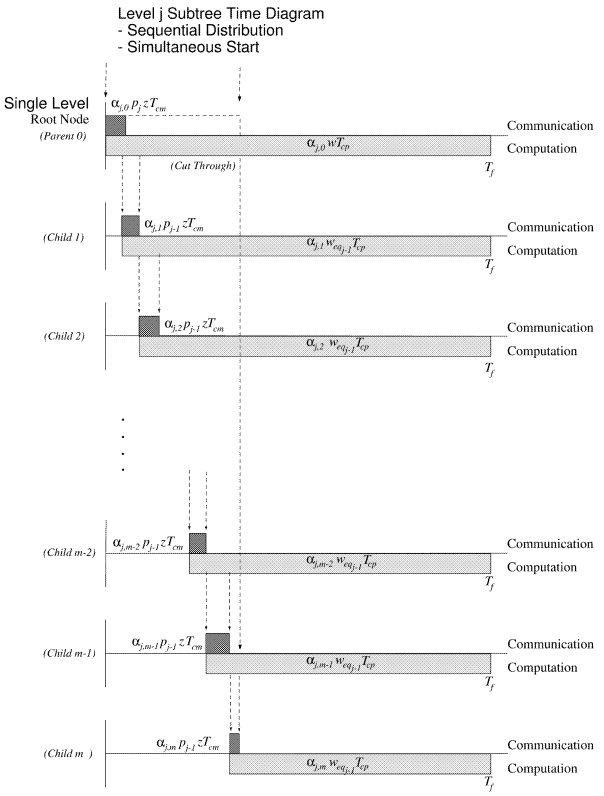


Fig. 5. Timing diagram of level j subtree with sequential distribution and simultaneous start.

$$= \frac{w}{w_{eq_{j-1}}} (1 - p_j \sigma) = \frac{1}{\gamma_{j-1}} (1 - p_j \sigma) \tag{33}$$

where $\sigma = z T_{cm} / w T_{cp}$.

It is also assumed that $w T_{cp} > p_j z T_{cm}$ (communication is faster than computing). Here (29)

$$\alpha_{j,i} = \frac{w_{eq_{j-1}} T_{cp} - p_{j-1} z T_{cm}}{w_{eq_{j-1}} T_{cp}} \alpha_{j,i-1} = q_{eq_{j-1}} \alpha_{j,i-1}$$

$$= \left[\prod_{l=2}^{i} (q_{eq_j - 1}) \right] \alpha_{j,1} = (q_{eq_{j-1}})^{i-1} \alpha_{j,1} \tag{34}$$

where $i = 2,3,\ldots,m$ and $q_{eq_{j-1}} = (w_{eq_{j-1}} T_{cp} - p_{j-1} z T_{cm}) / w_{eq_{j-1}} T_{cp}$. It is assumed as before that $w_{eq_{j-1}} T_{cp} > p_{j-1} z T_{cm}$. The expression of $q_{eq_{j-1}}$ can be manipulated as follows:

$$q_{eq_{j-1}} = 1 - \frac{w p_{j-1} z T_{cm}}{w_{eq_{j-1}} w T_{cp}} = 1 - \frac{w p_{j-1}}{w_{eq_{j-1}}} \sigma = 1 - \frac{p_{j-1}}{\gamma_{j-1}} \sigma \tag{35}$$

where $\sigma = z T_{cm} / w T_{cp}$ and $\gamma_{j-1} = w_{eq_{j-1}} / w$.

According to (31) and (34), the normalization equation (30) for the jth subtree leads to

$$\left\{ \frac{1}{k_{eq_{j-1}}} + 1 + \sum_{i=2}^{m} \left[\prod_{l=2}^{i} (q_{eq_{j-1}}) \right] \right\} \times \alpha_{j,1} = 1. \tag{36}$$

Hence the value of $\alpha_{j,1}$ is expressed as

$$\alpha_{j,1} = \frac{1}{\frac{1}{k_{eq_{j-1}}} + 1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{j-1}})\right]}. \quad (37)$$

Consequently, the equivalent finish time becomes

$$T_{f,m}^{h,j} = \alpha_{j,0} w T_{cp} = \frac{1}{k_{eq_{j-1}}}\alpha_{j,1} w T_{cp}$$
$$= \frac{1}{1 + k_{eq_{j-1}}\left\{1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{j-1}})\right]\right\}} w T_{cp}. \quad (38)$$

Since $w_{eq_j} T_{cp} = T_{f,m}^{h,j}$, (38) becomes

$$w_{eq_j} T_{cp} = \frac{1}{1 + k_{eq_{j-1}}\left\{1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{j-1}})\right]\right\}} w T_{cp}. \quad (39)$$

According to (34), (35), and (39), the expression of γ_j is derived as follows:

$$\gamma_j = \frac{w_{eq_j}}{w} = \frac{1}{1 + k_{eq_{j-1}}\left\{1 + \sum_{i=2}^{m}(q_{eq_{j-1}})^{i-1}\right\}}$$
$$= \frac{1}{1 + k_{eq_{j-1}}\left\{1 + \frac{q_{eq_{j-1}} - q_{eq_{j-1}}^{m}}{1 - q_{eq_{j-1}}}\right\}}$$
$$= \frac{1}{1 + \frac{1 - p_j\sigma}{p_{j-1}\sigma}\left\{1 - \left(1 - \frac{p_{j-1}}{\gamma_{j-1}}\sigma\right)^{m}\right\}}, \quad j = 1,2,\ldots,k-1. \quad (40)$$

Since the inverse computation capability of each node is w, it concludes that $w_{eq_0} = w$. Hence the initial value of γ_j is obtained as follows:

$$\gamma_0 = \frac{w_{eq_0}}{w} = \frac{w}{w} = 1. \quad (41)$$

For a homogeneous multilevel nonfat tree, the bandwidth of each transmission links is the same, that is, $p_j = 1$. So from (40)

$$\gamma_j = \frac{1}{1 + \frac{1-\sigma}{\sigma}\left\{1 - \left(1 - \frac{1}{\gamma_{j-1}}\sigma\right)^{m}\right\}}$$
$$= \frac{\sigma}{1 - (1-\sigma)\left(1 - \frac{\sigma}{\gamma_{j-1}}\right)^{m}}, \quad j = 1,2,\ldots,k-1. \quad (42)$$

According to (42) and $\gamma_0 = 1$, we can derive

$$\gamma_1 = \frac{\sigma}{1 - (1-\sigma)\left(1 - \frac{\sigma}{\gamma_0}\right)^{m}} = \frac{\sigma}{1 - (1-\sigma)^{m+1}}$$
$$\gamma_2 = \frac{\sigma}{1 - (1-\sigma)\left(1 - \frac{\sigma}{\gamma_1}\right)^{m}} = \frac{\sigma}{1 - (1-\sigma)^{m^2+m+1}}$$
$$\gamma_3 = \frac{\sigma}{1 - (1-\sigma)\left(1 - \frac{\sigma}{\gamma_2}\right)^{m}} = \frac{\sigma}{1 - (1-\sigma)^{m^3+m^2+m+1}}$$
$$= \frac{\sigma}{1 - (1-\sigma)^{\sum_{l=0}^{3} m^l}}$$
$$\vdots$$

Consequently, the general form of γ_j of nonfat tree is obtained as

$$\gamma_j = \frac{\sigma}{1 - (1-\sigma)^{\sum_{l=0}^{j} m^l}}, \quad j = 0,1,2,\ldots,k-1. \quad (43)$$

2) *Level k Subtree: Root Node with Data Storage*: In this subsection two types of distribution model for the topmost level subtree, level k, are discussed. One is sequential distribution, the other is simultaneous distribution. Generally simultaneous distribution requires a central processing unit (CPU) be fast enough to continually load all output buffers to its children. If the buffer capacity of the parent node cannot satisfy this basic requirement for simultaneous distribution, the sequential distribution model discussed in IIIB2a can be used. The simultaneous model is described in IIIB2b.

2a) *Level k subtree using sequential distribution*: In this part the expression for the topmost level subtree using sequential distribution is derived. The start strategy used here is simultaneous start.

The timing diagram of level k subtree using sequential distribution is illustrated in Fig. 6. According to this figure, the fundamental recursive equation can be obtained as follows:

$$\alpha_{k,0} w T_{cp} = \alpha_{k,1} w_{eq_{k-1}} T_{cp} \quad (44)$$
$$\alpha_{k,i-1} w_{eq_{k-1}} T_{cp} = \alpha_{k,i} w_{eq_{k-1}} T_{cp} + \alpha_{k,i-1} p_{k-1} z T_{cm}, \quad i = 2,3,\ldots,m. \quad (45)$$

The normalization equation for the topmost subtree is

$$\alpha_{k,0} + \alpha_{k,1} + \alpha_{k,2} + \cdots + \alpha_{k,m} = 1. \quad (46)$$

This gives $m + 1$ linear equations with $m + 1$ unknowns. Then from (44)

$$\alpha_{k,0} = \frac{w_{eq_{k-1}} T_{cp}}{w T_{cp}}\alpha_{k,1} = \frac{1}{k_{eq_{k-1}}}\alpha_{k,1}. \quad (47)$$

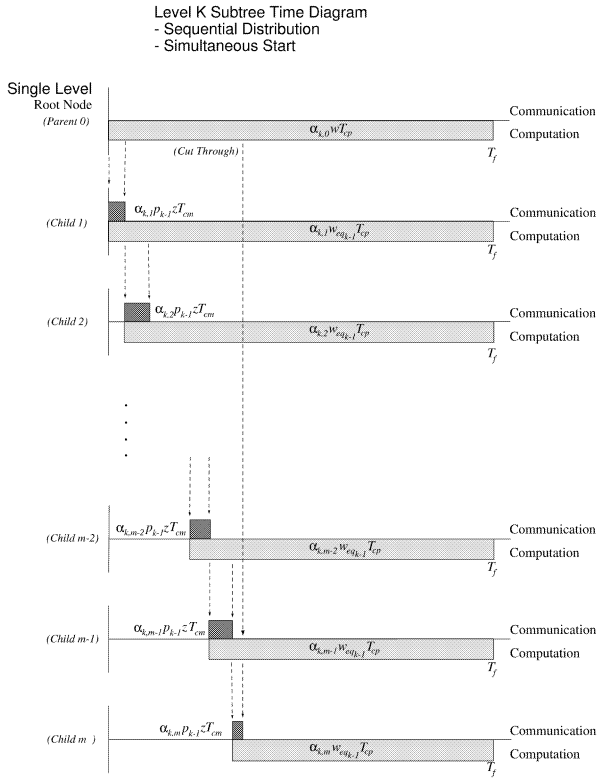


Fig. 6. Timing diagram of level k subtree using sequential distribution and simultaneous start.

Here, we let

$$k_{eq_{k-1}} = \frac{wT_{cp}}{w_{eq_{j-1}}T_{cp}} = \frac{w}{w_{eq_{j-1}}} = \frac{1}{\gamma_{k-1}}. \tag{48}$$

Now from (45)

$$\alpha_{k,i} = \frac{w_{eq_{k-1}}T_{cp} - p_{k-1}zT_{cm}}{w_{eq_{k-1}}T_{cp}}\alpha_{k,i-1} = q_{eq_{k-1}}\alpha_{k,i-1}$$

$$= \left[\prod_{l=2}^{i}(q_{eq_k-1})\right]\alpha_{k,1} \tag{49}$$

$$= (q_{eq_{k-1}})^{i-1}\alpha_{k,1}, \qquad i = 2,3,\ldots,m. \tag{50}$$

Let

$$q_{eq_{k-1}} = 1 - \frac{wp_{k-1}zT_{cm}}{w_{eq_{k-1}}wT_{cp}} = 1 - \frac{wp_{k-1}}{w_{eq_{k-1}}}\sigma = 1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma$$

$$\text{where} \quad \sigma = zT_{cm}/wT_{cp}. \tag{51}$$

According to (48) and (49), the normalization equation becomes

$$\left\{\frac{1}{k_{eq_{k-1}}} + 1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{k-1}})\right]\right\} \times \alpha_{k,1} = 1. \tag{52}$$

Consequently, the value of $\alpha_{k,1}$ can be obtained as follows:

$$\alpha_{k,1} = \frac{1}{\frac{1}{k_{eq_{k-1}}} + 1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{k-1}})\right]}. \tag{53}$$

The equivalent finish time $T_{f,m}^{h,k}$ is derived as

$$T_{f,m}^{h,k} = \alpha_{k,0}wT_{cp} = \frac{1}{k_{eq_{k-1}}}\alpha_{k,1}wT_{cp}$$

$$= \frac{1}{1 + k_{eq_{k-1}}\left\{1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{k-1}})\right]\right\}}wT_{cp}. \tag{54}$$

Because $w_{eq_k}T_{cp} = T_{f,m}^{h,k}$, (54) becomes

$$w_{eq_k}T_{cp} = \frac{1}{1 + k_{eq_{k-1}}\left\{1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{k-1}})\right]\right\}}wT_{cp}. \tag{55}$$

Thus

$$\frac{w_{eq_k}}{w} = \frac{1}{1 + k_{eq_{k-1}}\left\{1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{k-1}})\right]\right\}}. \tag{56}$$

According to Definition 1 in Section II and equations from (48), (50), and (51) we obtain the expression of γ_k as follows:

$$\gamma_k = \frac{w_{eq_k}}{w} = \frac{1}{1 + k_{eq_{k-1}}\left\{1 + \sum_{i=2}^{m}\left[\prod_{l=2}^{i}(q_{eq_{k-1}})\right]\right\}}$$

$$= \frac{1}{1 + k_{eq_{k-1}}\left\{1 + \sum_{i=2}^{m}(q_{eq_{k-1}})^{i-1}\right\}} \tag{57}$$

$$= \frac{1}{1 + \frac{1}{\gamma_{k-1}}\left\{1 + \frac{\left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right) - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)^{m}}{1 - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)}\right\}}$$

$$= \frac{1}{1 + \frac{1}{p_{k-1}\sigma}\left\{1 - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)^{m}\right\}}. \tag{58}$$

Then the speedup is

$$\text{Speedup} = 1 + \frac{1}{p_{k-1}\sigma}\left\{1 - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)^{m}\right\}. \tag{59}$$

1) For a homogeneous multilevel nonfat tree the bandwidth of each transmission links is the same, that is, $p_j = 1$. Then from (42), if $j = k-1$, then

$$\gamma_{k-1} = \frac{\sigma}{1 - (1-\sigma)^{\sum_{l=0}^{k-1}m^l}}. \tag{60}$$

Hence, (58) can be solved as follows:

$$\gamma_k = \frac{1}{1 + \frac{1}{\sigma}\left\{1 - \left(1 - \frac{\sigma}{\gamma_{k-1}}\right)^{m}\right\}}$$

$$= \frac{1}{1 + \frac{1}{\sigma}\{1 - (1-\sigma)^{\sum_{l=1}^{k}m^l}\}}. \tag{61}$$

This leads to a closed solution and the speedup of the multilevel nonfat tree is

$$\text{Speedup} = \frac{1}{\gamma_k} = 1 + \frac{1}{\sigma}\{1-(1-\sigma)^{\sum_{l=1}^{k} m^l}\}$$

$$\text{where} \quad \gamma_0 = 1. \tag{62}$$

2) For a homogeneous multilevel nonfat tree, the expressions of γs are as follows:

$$\gamma_0 = 1 \tag{63}$$

$$\gamma_j = \frac{\sigma}{1-(1-\sigma)\left(1-\dfrac{\sigma}{\gamma_{j-1}}\right)^m}$$

$$\text{where} \quad j = 1,2,\ldots,k-1 \tag{64}$$

$$\gamma_k = \frac{1}{1+\dfrac{1}{\sigma}\{1-(1-\sigma)^{\sum_{l=1}^{k} m^l}\}}. \tag{65}$$

Consequently, the speedup is expressed as

$$\text{Speedup} = 1 + \frac{1}{\sigma}\{1-(1-\sigma)^{\sum_{l=1}^{k} m^l}\}. \tag{66}$$

2b) *Level k subtree using simultaneous distribution*: In the topmost level subtree, the root of this level is the topmost root for the entire tree. Since all the data for distribution in the topmost root is already stored in this node, it is not necessary for this node to wait for data to come in from its parent (if the parent exists) under cut through transmission. Therefore, unlike the rest of the levels below level k, in a nonfat tree, the topmost level k can use simultaneous distribution instead of sequential distribution to improve the performance.

Using simultaneous distribution, the top equivalent subtree timing, level k, is illustrated in Fig. 7. According to the Fig. 7, the fundamental recursive equations are

$$\alpha_{k,0} w T_{cp} = \alpha_{k,1} w_{eq_{k-1}} T_{cp} \tag{67}$$

$$\alpha_{k,i-1} w_{eq_{k-1}} T_{cp} = \alpha_{k,i} w_{eq_{k-1}} T_{cp}, \qquad i = 2,3,\ldots,m. \tag{68}$$

In addition, the normalization equation for the single level subtree with intelligent root (that can process load as well as distribute it) is

$$\alpha_{k,0} + \alpha_{k,1} + \alpha_{k,2} + \cdots + \alpha_{k,m} = 1. \tag{69}$$

This gives $m+1$ linear equations with $m+1$ unknowns. These equations can be solved recursively in the same manner as was done in the previous sections to obtain

$$k_{eq_{k-1}} = \frac{w T_{cp}}{w_{eq_{j-1}} T_{cp}} = \frac{w}{w_{eq_{j-1}}} = \frac{1}{\gamma_{k-1}} \tag{70}$$

$$\gamma_k = \frac{w_{eq_k}}{w} = \frac{1}{1 + k_{eq_{k-1}} \times m} = \frac{1}{1+\dfrac{m}{\gamma_{k-1}}}. \tag{71}$$

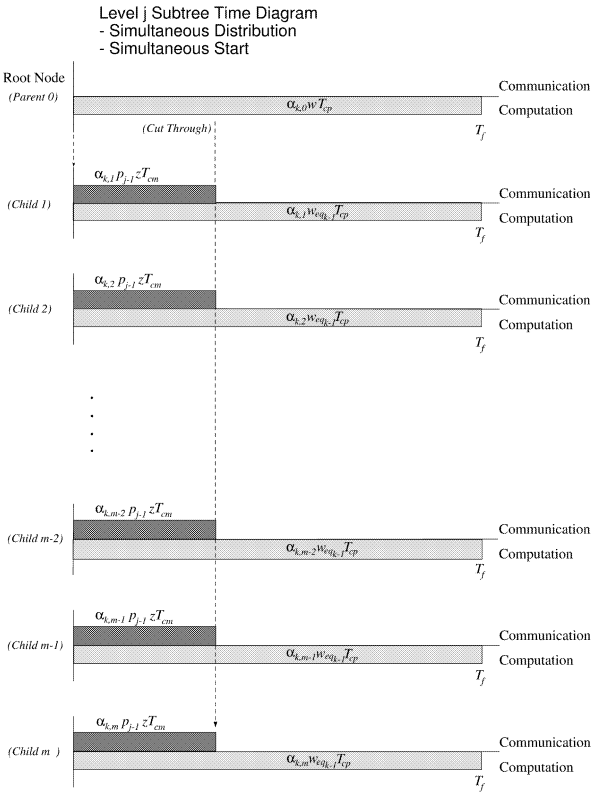


Fig. 7. Timing diagram of level k subtree using simultaneous distribution and simultaneous start.

The speedup of this multilevel tree using simultaneous distribution in the topmost level subtree is

$$\text{Speedup} = \frac{1}{\gamma_k} = 1 + \frac{m}{\gamma_{k-1}}. \tag{72}$$

1) Consider a homogeneous multilevel nonfat tree, which uses simultaneous distribution in the topmost level subtree but sequential distribution in the levels below the topmost level. (Nonfat tree means that all the bandwidth of each transmission links is the same, $p_j = 1$).

Now from (43), we obtain

$$\gamma_{k-1} = \frac{\sigma}{1-(1-\sigma)^{\sum_{l=0}^{k-1} m^l}}. \tag{73}$$

Hence the value of γ_k can be obtained as follows:

$$\gamma_k = \frac{1}{1+\dfrac{m}{\gamma_{k-1}}} = \frac{1}{1+m\Big/\dfrac{\sigma}{1-(1-\sigma)^{\sum_{l=0}^{k-1} m^l}}}$$

$$= \frac{1}{1+\dfrac{m}{\sigma}[1-(1-\sigma)^{\sum_{l=0}^{k-1} m^l}]}. \tag{74}$$

Finally, the speedup of a multilevel nonfat tree using sequential distribution but simultaneous distribution at the topmost level is

$$\text{Speedup} = \frac{1}{\gamma_k} = 1 + \frac{m}{\sigma}[1-(1-\sigma)^{\sum_{l=0}^{k-1} m^l}]. \tag{75}$$

2) For a homogeneous multilevel fat tree, the values of γs are as follows:

$$\gamma_0 = 0$$

$$\gamma_j = \frac{1}{1 + \frac{1 - p_j\sigma}{p_{j-1}\sigma}\left\{1 - \left(1 - \frac{p_{j-1}}{\gamma_{j-1}}\sigma\right)^m\right\}} \quad (76)$$

$$\gamma_k = \frac{1}{1 + \frac{m}{\gamma_{k-1}}}. \quad (77)$$

The speedup is

$$\text{Speedup} = 1 + \frac{m}{\gamma_{k-1}}. \quad (78)$$

C. Speedup Calculation for Heterogeneous Trees

For tractability and to produce recursive expressions, Section IIIA and Section IIIB assumed a homogeneous symmetrical tree. Speedup can be calculated for heterogeneous and nonsymmetrical trees by collapsing single level subtrees, starting from the bottom of the multilevel tree and working upwards, until a single equivalent processor representing the operating characteristics of the entire tree is found [2, 21]. Using this, the calculation of speedup relative to a reference processor is straight forward. Since relatively simple recursive algebraic expressions are not possible, this procedure is best done recursively by a computer program.

D. Use of Multi-Installment Scheduling

A known technique for decreasing the time processors wait to receive load under sequential distribution is to distribute load sequentially and periodically in small installments or rounds [7, 18]. Multi-installment scheduling can be used in conjunction with the optimal cut through switching presented here to boost speedup. If communication speeds are faster than computation speeds (the assumption in here) then as installment size shrinks a (saturating) performance improvement results. If communication speeds are relatively slower than computation speeds (as is the case in some wireless networks) "gaps" will result in the timing and the equations presented here will not be valid. In this case though the use of store and forward switching will result in excessive store and forward delay. The use of cut through switching, by comparison, will lead to a dramatic performance improvement.

IV. SEQUENTIAL DISTRIBUTION USING STORE AND FORWARD SWITCHING

Under store and forward switching, a node must completely receive the load for itself and its descendants before beginning to compute and distribute load to its children.

Again, for the purposes of determining the optimal load allocations, the single level trees within the overall multilevel tree are divided into the root single level subtree (level k) and the single level subtrees below the root subtree (level $1,2,\ldots,k-1$). It is assumed that all data is available (and stored) at the root at $t = 0$. Thus the root can immediately deliver load to its children at level k. This is the root node with data storage case.

For the other single level trees, it is assumed that its load must be completely received by a single level tree root before being distributed to its children. After this, load is relayed through the root to its children in store and forward mode. This is the root node without data storage case.

In both cases, only the simultaneous start strategy is considered. In the simultaneous start strategy each processor begins processing the received data while it continues to receive the data. In the following both cases are examined, first in the context of single level trees in isolation and then in the context of multilevel trees.

A. Processors with Sequential Distribution. Homogeneous Multilevel Fat Tree Analysis

A fat tree architecture is now considered where upper links have more capacity than lower links in such a way that each node has bandwidth $1/z$ to the root.

We proceed by aggregating single level subtrees into equivalent processors, starting from the bottom of the tree and working upwards [21]. The tree's bottommost single level subtrees are at level 1, and the tree's topmost (including the root) subtree is at level k.

Consider a homogeneous multilevel fat tree network where all processors have the same inverse computing speed w, and all links of level j have the same inverse transmission speed $p_{j-1}z$. We use the same tree labeling as in Fig. 2. The value of p_{j-1} is defined in Definition 3 from Section II, that is,

$$p_{j-1}z = \left[\left(\sum_{l=0}^{j-1} m^l\right)^{-1}\right] z. \quad (79)$$

Again, the process for the load distribution of a multilevel fat tree network using the store and forward switching for computing and communicating from upper level to lower level can be represented by Gantt chart-like timing diagram.

We derive the speedup of the whole multilevel tree by successively collapsing single level trees into

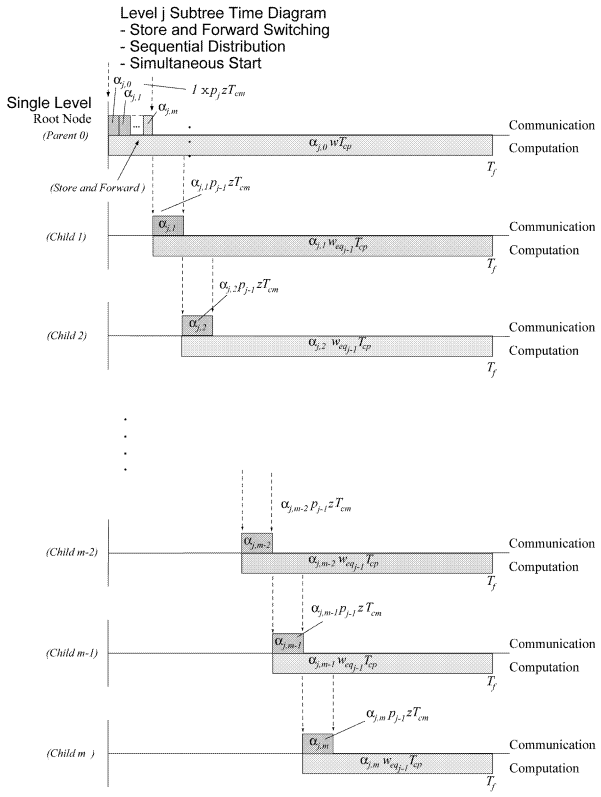


Fig. 8. Timing diagram of level j subtree using store and forward switching, sequential distribution, and simultaneous start for root node without data storage.

equivalent nodes until the entire tree is collapsed into an equivalent node. We first use the root without data storage model for levels ($j = 1,2,\ldots,k-1$), and then use the root with data storage model for the top level (level k).

B. Level j Subtree. Root Node without Data Storage

The Gantt chart-like timing diagram for jth level subtree is illustrated in Fig. 8. According to Fig. 8, the fundamental recursive equations of the jth level tree network are

$$\alpha_{j,0} w T_{cp} = \alpha_{j,1} w_{eq_{j-1}} T_{cp} + 1 \times p_j z T_{cm} \tag{80}$$

$$\alpha_{j,i-1} w_{eq_{j-1}} T_{cp} = \alpha_{j,i} w_{eq_{j-1}} T_{cp} + \alpha_{j,i-1} p_{j-1} z T_{cm}, \quad i = 2,3,\ldots,m. \tag{81}$$

Here as we move up the tree, collapsing single level trees into equivalent processors, the single level trees consist of a root with inverse speed w and children nodes of inverse speed $w_{eq_{j-1}}$. The normalization equation for the jth single level tree with intelligent root is

$$\alpha_{j,0} + \alpha_{j,1} + \alpha_{j,2} + \cdots + \alpha_{j,m} = 1. \tag{82}$$

This yields $m+1$ linear equations with $m+1$ unknowns. Now using (80),

$$\alpha_{j,0} = \frac{w_{eq_{j-1}} T_{cp} \alpha_{j,1} + p_j z T_{cm}}{w T_{cp}} = \frac{1}{k_{eq_{j-1}}} \alpha_{j,1} + p_j \sigma \tag{83}$$

where

$$k_{eq_{j-1}} = w / w_{eq_{j-1}} = 1/\gamma_{j-1} \qquad \text{and} \qquad \sigma = z T_{cm} / w T_{cp}. \tag{84}$$

Now from (81),

$$\alpha_{j,i} = \frac{w_{eq_{j-1}} T_{cp} - p_{j-1} z T_{cm}}{w_{eq_{j-1}} T_{cp}} \alpha_{j,i-1} = q_{eq_{j-1}} \alpha_{j,i-1}$$

$$= \left[\prod_{l=2}^{i} q_{eq_j - 1} \right] \alpha_{j,1} = (q_{eq_{j-1}})^{i-1} \alpha_{j,1} \tag{85}$$

where $i = 2,3,\ldots,m$.

Naturally $w_{eq_{j-1}} T_{cp} > p_{j-1} z T_{cm}$ as communication time is assumed as be faster than computation time. We note that

$$q_{eq_{j-1}} = \frac{w_{eq_{j-1}} T_{cp} - p_{j-1} z T_{cm}}{w_{eq_{j-1}} T_{cp}} = 1 - \frac{p_{j-1} z T_{cm}}{w_{eq_{j-1}} T_{cp}} \times \frac{w}{w}$$

$$= 1 - \frac{p_{j-1}}{\gamma_{j-1}} \times \sigma. \tag{86}$$

Consequently,

$$\prod_{l=2}^{i} q_{eq_j - 1} = \left(1 - \frac{p_{j-1}}{\gamma_{j-1}} \times \sigma \right)^{i-1}. \tag{87}$$

According to (83) and (85), the normalization equation (82) becomes

$$p_j \sigma + \frac{1}{k_{eq_{j-1}}} \alpha_{j,1} + \alpha_{j,1} + \sum_{i=2}^{m} \alpha_{j,i} = 1$$

$$\left\{ \frac{1}{k_{eq_{j-1}}} + 1 + \sum_{i=2}^{m} \left[\prod_{l=2}^{i} q_{eq_{j-1}} \right] \right\} \times \alpha_{j,1} = 1 - p_j \sigma.$$

Finally, one obtains the value of $\alpha_{j,1}$:

$$\alpha_{j,1} = \frac{1 - p_j \sigma}{\frac{1}{k_{eq_{j-1}}} + 1 + \sum_{i=2}^{m} \left[\prod_{l=2}^{i} q_{eq_{j-1}} \right]}. \tag{88}$$

Proceeding as in the previous section, one can find γ_j, the inverse of speedup as

$$\gamma_j = \frac{w_{eq_j}}{w} = \frac{p_{j-1}\sigma + p_j \sigma \left[1 - \left(1 - \frac{p_{j-1}}{\gamma_{j-1}} \sigma \right)^m \right]}{p_{j-1}\sigma + \left[1 - \left(1 - \frac{p_{j-1}}{\gamma_{j-1}} \sigma \right)^m \right]}. \tag{89}$$

C. Level k Subtree. Root Node with Data Storage

In this subsection two types of distribution models for the topmost level subtree, level k, are discussed. One is sequential distribution, the other is simultaneous distribution. Generally simultaneous distribution requires a CPU be fast enough to continually load all output buffers to its children. According to the specification as above, the timing diagram and recursive formulae for speedup are the same as Section IIIB2.

1) *Level k Subtree Using Sequential Distribution*: The timing diagram of level k subtree using sequential distribution is the same as illustrated in Fig. 6. According to Fig. 6, the solution of γ_k and speedup are obtained as (58) and (59) as follows:

$$\gamma_k = \frac{1}{1 + \frac{1}{p_{k-1}\sigma}\left\{1 - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)^m\right\}} \tag{90}$$

$$\text{Speedup} = 1 + \frac{1}{p_{k-1}\sigma}\left\{1 - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)^m\right\}. \tag{91}$$

2) *Level k Subtree Using Simultaneous Distribution*: The timing diagram of level k subtree using sequential distribution is the same as illustrated in Fig. 7. According to Fig. 7, the solution of γ_k and speedup are obtained as (92) and (93) as follows:

$$\gamma_k = \frac{1}{1 + \frac{m}{\gamma_{k-1}}} \tag{92}$$

$$\text{Speedup} = \frac{1}{\gamma_k}. \tag{93}$$

V. SUMMARY AND PERFORMANCE EVALUATION

For cut through switching and store and forward switching the recursive speedup formulae are developed as above and summarized as follows.

A. Homogeneous Multilevel Fat Tree

In this part, we summarize the recursive formulae for a multilevel fat tree using sequential distribution under cut through switching and store and forward switching.

For level j:

$$\gamma_0 = 1 \tag{94}$$

$$\gamma_j \text{ (cut through)} = \frac{1}{1 + \frac{1 - p_j\sigma}{p_{j-1}\sigma}\left\{1 - \left(1 - \frac{p_{j-1}}{\gamma_{j-1}}\sigma\right)^m\right\}} \tag{95}$$

$$\gamma_j \text{ (store and forward)} = \frac{p_{j-1}\sigma + p_j\sigma\left[1 - \left(1 - \frac{p_{j-1}}{\gamma_{j-1}}\sigma\right)^m\right]}{p_{j-1}\sigma + \left[1 - \left(1 - \frac{p_{j-1}}{\gamma_{j-1}}\sigma\right)^m\right]} \tag{96}$$

where $j = 1,2,\ldots,k-1$.

1) If the distribution of the k level is sequential,

$$\gamma_k = \frac{1}{1 + \frac{1}{p_{k-1}\sigma}\left\{1 - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)^m\right\}}. \tag{97}$$

The speedup is

$$\text{Speedup} = \frac{1}{\gamma_k} = 1 + \frac{1}{p_{k-1}\sigma}\left\{1 - \left(1 - \frac{p_{k-1}}{\gamma_{k-1}}\sigma\right)^m\right\}. \tag{98}$$

2) If the distribution of the k level is simultaneous,

$$\gamma_k = \frac{1}{1 + \frac{m}{\gamma_{k-1}}}. \tag{99}$$

The speedup is

$$\text{Speedup} = \frac{1}{\gamma_k} = 1 + \frac{m}{\gamma_{k-1}}. \tag{100}$$

B. Homogeneous Multilevel Nonfat Tree

The homogeneous multilevel nonfat tree using cut through switching and store and forward switching under the sequential distribution and simultaneous start and method assume all the bandwidth of each transmission link is the same, $p_j = 1$. This is one special case of the homogeneous multilevel fat tree. The formulae of the tree using cut through switching can be obtained as closed-form formulae. The following formulae apply only to the model using cut through switching.

$$\gamma_0 = 1 \tag{101}$$

$$\gamma_j = \frac{\sigma}{1 - (1 - \sigma)^{\sum_{l=0}^{j} m^l}}, \qquad j = 1,2,\ldots,k-1. \tag{102}$$

1) If the distribution of the k level is sequential

$$\gamma_k = \frac{1}{1 + \frac{1}{\sigma}\{1 - (1 - \sigma)^{\sum_{l=1}^{k} m^l}\}}. \tag{103}$$

The speedup is

$$\text{Speedup} = \frac{1}{\gamma_k} = 1 + \frac{1}{\sigma}\{1 - (1 - \sigma)^{\sum_{l=1}^{k} m^l}\}. \tag{104}$$

2) If the distribution of the k level is simultaneous,

$$\gamma_k = \frac{1}{1 + \frac{m}{\sigma}[1 - (1 - \sigma)^{\sum_{l=0}^{k-1} m^l}]}. \tag{105}$$

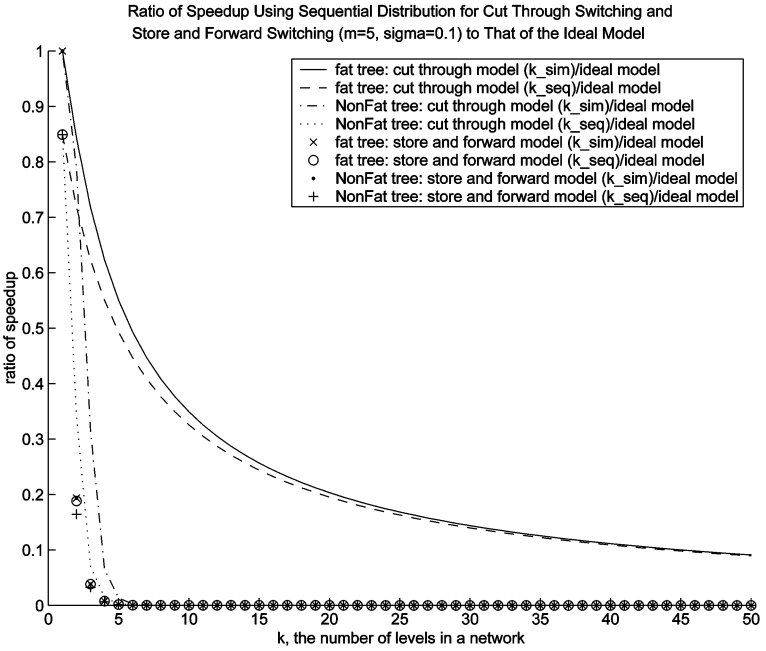


Fig. 9. Ratio of speedup for multilevel tree models using simultaneous start or staggered start to that for an ideal model.

The speedup is

$$\text{Speedup} = \frac{1}{\gamma_k} = 1 + \frac{m}{\sigma}[1 - (1-\sigma)^{\sum_{l=0}^{k-1} m^l}]. \tag{106}$$

C. Performance Evaluation

According to the recursive equation (95), (96), (97), and (99) for the fat tree model and setting $p_j = 1$ (where $j = 0, 1, \ldots, k-1$) for the nonfat tree model, we obtain the ratio of the speedup for these eight cases to the speedup of the ideal model and then illustrate the result in Fig. 9. The ideal model has extremely fast communication time.

As shown in Fig. 9, the ratio of the speedup of the store and forward models for the fat tree and nonfat tree networks to that of the ideal model approaches zero very quickly as the number of tree levels is increased. This means that even if the store and forward model uses a fat tree network, the speedup saturates quickly under the sequential distribution. The cut through model has the best performance with a fat tree network. Even this model with a nonfat tree network also has better performance than that of a store and forward model with a fat tree network.

VI. CONCLUSION

The most important results of this paper are simple recursive solutions for speedup and solution time for a divisible load optimally scheduled on a multilevel tree with virtual cut through switching. This is done for a variety of scheduling features under a number of scenarios. This work is more general than the exact situations discussed here and the methodology can be applied to a wide variety of load distribution scheduling policies.

Aerospace applications will certainly see the increasing use of multiple sensor/multiple processor systems. In such systems the ability to do processing in a solution time optimal manner decreases response time and minimizes the amount of hardware necessary to accomplish a task. With the increasing ubiquity and decreasing cost of such systems, this tractable performance evaluation approach should be of interest.

ACKNOWLEDGMENTS

The assistance of M. Moges in preparing this article is appreciated.

REFERENCES

[1] Robertazzi, T. G. (2003)
Ten reasons to use divisible load theory.
Computer, **36**, 5 (2003), 63–68.

[2] Bharadwaj, V., Ghose, D., Mani, V., and Robertazzi, T. G. (1996)
Scheduling Divisible Loads in Parallel and Distributed Systems.
Los Alamitos, CA: IEEE Computer Society Press, 1996.

[3] Bharadwaj, V., Ghose, D., and Robertazzi, T. G. (2003)
A new paradigm for load scheduling in distributed systems.
Special issue of *Cluster Computing* on Divisible Load Scheduling, **6**, 1 (2003), 7–18.

[4] Cheng, Y. C., and Robertazzi, T. G. (1988)
Distributed computation with communication delays.
IEEE Transactions on Aerospace and Electronic Systems, **24**, 6 (1988), 700–712.

[5] Barlas, G. D. (1998)
Collection aware optimum sequencing of operations and closed form solutions for the distribution of divisible load on arbitrary processor trees.
IEEE Transactions on Parallel and Distributed Systems, **9**, 5 (1998), 429–441.

[6] Bataineh, S., and Robertazzi, T. G. (1991)
Bus oriented load sharing for a network of sensor driven processors.
IEEE Transactions on Systems, Man and Cybernetics, **21**, 5 (1991), 1202–1205.

[7] Bharadwaj, V, Ghose, D., and Mani, V. (1995)
Multi-installment load distribution in tree networks with delay.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 2 (1995), 555–567.

[8] Bharawaj, V., Ghose, D., and Mani, V. (1995)
An efficient load distribution strategy for a distributed linear network of processors with communications delays.
Computers and Mathematics with Applications, **29**, 9 (1995), 95–112.

[9] Blazewicz, J., and Drozdowski, M. (1995)
Scheduling divisible jobs on hypercubes.
Parallel Computing, **21**, 12 (1995), 1945–1956.

[10] Blazewicz, J., and Drozdowski, M. (1996)
The performance limits of a two dimensional network of load sharing processors.
Foundations of Computing and Decision Sciences, **21**, 1 (1996), 3–15.

[11] Blazewicz, J., and Drozdowski, M. (1997)
Distributed processing of divisible jobs with communication start-up costs.
Discrete Applied Mathematics, **76**, 1–3 (1977), 21–41.

[12] Blazewicz, J., Drozdowski, M., Guinand, F., and Trystram, D. (1999)
Scheduling a divisible task in a 2-dimensional mesh.
Discrete Applied Mathematics, **94** (1999), 35–50.

[13] Drozdowski, M., and Glazek, W. (1999)
Scheduling a divisible load in a three-dimensional mesh of processors.
Parallel Computing, **25** (1999), 381–404.

[14] Kim, H. J., Jee, J. I., and Lee, J. G. (1996)
Optimal load distribution for tree network processors.
IEEE Transactions on Aerospace and Electronic Systems, **32**, 2 (1996), 607–612.

[15] Sohn, J., and Robertazzi, T. G. (1998)
Optimal time varying load sharing for divisible loads.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 3 (1998), 907–924.

[16] Bharadwaj, V., and Barlas, G. (2002)
Efficient scheduling strategies for processing multiple divisible loads on bus networks.
Journal of Parallel and Distributed Computing, **62** (2002), 132–151.

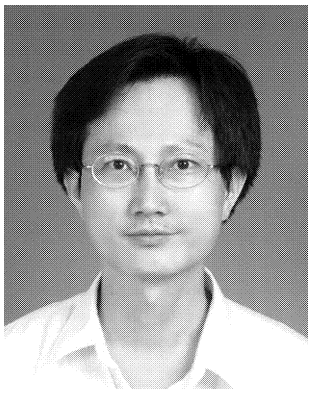
[17] Beaumont, O., Carter, L., Ferrante, J., Legrand, A., and Robert, Y. (2002)
Bandwidth-centric allocation of independent tasks on heterogeneous platforms.
In *Proceedings of the International Parallel and Distributed Processing Symposium* (IPDPS'02), 2002.

[18] Yang, Y., and Casanova, H. (2003)
UMR: A multi-round algorithm for scheduling divisible workloads.
In *Proceedings of the International Parallel and Distributed Processing Symposium* (IPDPS'03), 2003.

[19] Hung, J. T., Kim, H. J., and Robertazzi, T. G. (2002)
Scalable scheduling in parallel processors.
In *Proceedings of the 2002 Conference on Information Sciences and Systems*, Princeton University, Princeton, NJ, 2002.

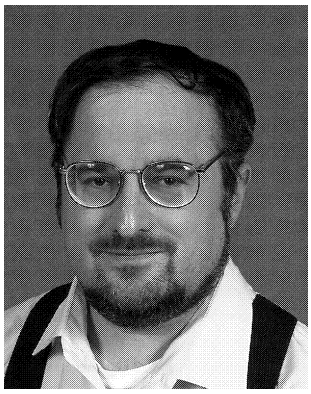
[20] Kim, H. J. (2003)
A novel load distribution algorithm for divisible loads.
Special issue of *Cluster Computing* on Divisible Load Scheduling, **6**, 1 (2003), 41–46.

[21] Cheng, Y. C., and Robertazzi, T. G. (1990)
Distributed computation for tree networks with communication delays.
IEEE Transactions on Aerospace and Electronic Systems, **26** (1990), 511–516.



Jui Tsun Hung received the B.S. and M.S. degrees in mechanical engineering from the National Sun Yat-Sen University and Chuang Yuan Christian University, Taiwan, in 1986 and 1991. He also received the M.S. and Ph.D. degrees in electrical engineering from the State University of New York at Stony Brook, NY, in 2001, and 2003, respectively.

He was a lecturer in the Wu Feng Institute of Technology and Commerce, Taiwan, from 1991 to 1997. In 1998, he joined Golden Circuit Electronics Corporation, Taoyuan, Taiwan, and worked on manufacturing printed circuit boards. He joined Wintek Corporation, Taichung, Taiwan, in 1999, as an R&D electrical engineer, where he was primarily engaged in LCD driver circuit modular design. He is currently a senior engineer in Memes Technology Corporation, Nangkang Software Park, Taipei, Taiwan, where his research interests are RF/microwave circuit design for wireless applications and scheduling theory for general tree models.



Thomas G. Robertazzi (S'75—M'77—SM'91) received the Ph.D. from Princeton University, Princeton, NJ, in 1981 and the B.E.E. from the Cooper Union, New York, NY in 1977.

He is presently a professor in the Department of Electrical and Computer Engineering at Stony Brook University, Stony Brook NY. In supervising a very active research area, he has published extensively in the areas of parallel processor and grid scheduling, ad hoc radio networks, telecommunications network planning, ATM switching, queueing, and Petri networks.

Dr. Robertazzi has authored, coauthored or edited four books in the areas of performance evaluation, scheduling and network planning.