1990

Load Balanced Tree Embeddings

Ajay K. Gupta

Susanne E. Hambruch
Purdue University, seh@cs.purdue.edu

Report Number:
90-957

Gupta, Ajay K. and Hambruch, Susanne E., "Load Balanced Tree Embeddings" (1990). *Department of Computer Science Technical Reports*. Paper 811.
<https://docs.lib.purdue.edu/cstech/811>

LOAD BALANCED TREE EMBEDDINGS

Ajay K. Gupta
Susanne E. Hambrusch

CSD-TR-957
February 1990

Load Balanced Tree Embeddings

Ajay K. Gupta ¹

Department of Computer Science
Western Michigan University
Kalamazoo, Michigan.

Susanne E. Hambruch ²

Department of Computer Sciences
Purdue University
West Lafayette, Indiana.

¹Research supported in part by ONR under contract N00014-86-K-0689 and by NSF under Grant MIP-87-15652.

²Research supported in part by ONR under contracts N00014-84-K-0502 and N00014-86-K-0689, and by NSF under Grant MIP-87-15652.

Abstract

The problem of embedding an n -processor architecture T into an m -processor architecture H for $n > m$ arises when algorithms designed for architectures of an ideal size are simulated on existing architectures which are of a fixed size. In this paper we present solutions to this embedding problem for the case when both architectures are complete binary trees and the embeddings are to achieve a balanced load. An embedding achieves a balanced load if every processor of H simulates at most $\lceil n/m \rceil$ processors of T . We show that in this case T can be embedded into H with a dilation of 1 and a congestion of $\min\{\lceil \frac{n}{m} \rceil, 2 \log n\}$. We also consider embeddings that achieve a balanced l/i load; i.e., every processor of H simulates at most $\lceil \frac{n+1}{2m} \rceil$ leaves and at most $\lceil \frac{n-1}{2m} \rceil$ interior processors of T . We present an embedding that achieves a balanced l/i load, a dilation of $2\lceil \log \log m \rceil + 1$ and a congestion of $O(\log n)$. We show that every embedding strategy achieving a balanced l/i load for all values of n and m must achieve a dilation of at least 3. We also consider the embedding problem when the edges of T have weights associated that are to be taken into account by the congestion.

1 Introduction

One of the major problems in parallel computation is the efficient reuse of parallel software. The architecture of a parallel machine has a significant influence on the choice of the problem solving and communication techniques. Furthermore, parallel algorithms are usually designed in terms of the input size of the problem and not in terms of the number of processors available on a host machine. Thus, it is difficult to port a program from one machine or an ideal environment to a machine of different architecture and/or size. In an attempt to attack the problem of portability of parallel algorithms, understanding the relationships between different architectures and identical architectures of different sizes is of vital importance.

In this paper we consider the problem of mapping an algorithm or a communication structure designed for a complete binary tree T of size n to that of a complete binary tree H of size m with $n > m$. We phrase the problem as a graph embedding problem. The concept of graph embeddings has proven to be a successful one in understanding relationships between interconnection networks [2, 4, 10, 11, 12, 15]. When $n > m$, one processor of H simulates a number of processors of T and the load of the processors of H becomes a crucial quantity. We concentrate on embeddings that achieve a balanced load; i.e., every processor of H simulates at most $\lceil \frac{n}{m} \rceil$ processors of T . Embeddings achieving a balanced load are of practical importance, since they make every processor of H share an equal load and in existing parallel machines every processor has a fixed capacity. We investigate two different types of balanced load. We show that while one type of balanced load can be achieved without an increase in other cost measures of an embedding, the second type increases the dilation. Before describing our results in detail, we give the necessary definitions.

When the architectures of both the machines are viewed as graphs, an embedding $\langle f, g \rangle$ of T into H is defined by a surjective mapping f from the processors of T to the processors of H together with a mapping g that maps every edge $e = (v, w)$ of T onto a path $g(e)$ connecting $f(v)$ and $f(w)$. We refer to f as the assignment. Two commonly and extensively studied cost measures of an embedding are the dilation and the congestion [2, 4, 10, 11, 15]. The *dilation* δ is defined as the maximum distance in H

between two adjacent processors in T , and the *congestion* λ is defined as the maximum number of paths over an edge in H , where every path represents an edge in T . If the edges of T have weights associated with them, we consider the *weighted congestion* λ_w . It is defined as the maximum over the sum of weights over an edge in H , where the weights are the weights of the edges in T . The *load* μ is defined as the maximum number of processors of T assigned to any processor of H . We say that an embedding achieves a *balanced load* when $\mu = \lceil \frac{n}{m} \rceil$. A more restricted type of balanced load is that of a balanced l/i load. An embedding achieves a balanced l/i load if every processor of H is assigned at most $\mu_l = \lceil \frac{n+1}{2m} \rceil$ leaves of T and at most $\mu_i = \lceil \frac{n-1}{2m} \rceil$ interior processors of T . The motivation for considering a balanced l/i load comes from the fact that a number of algorithms designed for tree networks require the leaf processors to have capabilities different from the ones of the interior processors [1, 3]. In fact, tree networks with this property have been designed [5, 14]. In addition, the question of whether embeddings with a balanced l/i load require different embedding strategies is interesting in its own right.

Embeddings of a network G into a host network H of the same topology, but smaller size have previously been studied in [6, 7]. For the case of complete binary trees when $m = \sqrt{n+1} - 1$ Fishburn and Finkel [7] present an embedding that gives $\delta = 1$, $\lambda = \mu = \Theta(n/m)$. Berman and Snyder [6] present embeddings by performing contractions which guarantee a dilation of 1, but do not achieve a balanced load. More recently, Sang and Sudborough [17] have investigated the problem of achieving a balanced load for meshes.

Our main results with respect to embeddings are Embedding 1 and Embedding 2 which are described in Sections 3 and 4, respectively. Embedding 1 achieves a balanced load, an optimal dilation of 1, and a congestion of $\min\{\lceil \frac{n}{m} \rceil, 2 \log n\}$. However, it has an unbalanced l/i load. Embedding 2, which achieves a balanced l/i load, has a dilation of $2\lceil \log \log m \rceil + 1$, and a congestion of $O(\log n)$. In both embeddings, every processor of H can easily compute its assignment in $O(\lceil \frac{n}{m} \rceil + \log m)$ time. Our embeddings make use of a number of interesting embedding strategies. We keep the load balanced and the congestion small by assigning subtrees of T to the processors of H . However, in order to obtain small dilation, the parents of the roots of these subtrees have to be assigned to nearby processors of H . Our overall strategy is to first obtain an initial unbalanced em-

bedding with small dilation and congestion, and to then refine this embedding carefully so that a balanced load is achieved along with small dilation and congestion. Obtaining a balanced $1/i$ load and keeping the cost measures small is considerably harder than just obtaining a balanced load. In fact, we show that the dilation must be at least 3 in an embedding strategy that achieves a balanced $1/i$ load for all values of n and m .

In Section 5 we consider embeddings for the case when the edges of T have weights associated with them. We show that it is NP-hard to achieve a minimum weighted congestion when the edges of T have arbitrary weights. We then consider *exponentially decreasing* weights on the edges of T . The edges of T have exponentially decreasing weights when every edge between a processor at level i and a processor at level $i+1$ in T has weight $w/2^i$, for some $w = 2^r > 0$ and $0 \leq i \leq \lceil \log n \rceil - 2$. Weights of this type arise, for example, when T models the amount of data to be transferred between processes in a divide-and-conquer strategy [16]. We prove an achievable lower bound of $\lambda_w = \Omega(w)$ for any embedding with a balanced load and exponentially decreasing weights.

2 Preliminaries

In this section we describe three embeddings that will be used throughout the paper: the isomorphic embedding, the shrink embedding, and the Different Level Subtree embedding. Let T' be an n' -node complete binary tree of height $k' - 1$ and H' be an m' -node complete binary tree of height $l' - 1$. For clarity reasons, we refer to nodes of H' as processing elements (PEs) and nodes of T' simply as nodes. Let $t'(i, j)$ (resp. $h'(i, j)$) denote the j^{th} leftmost node (resp. PE) on level i in T' (resp. H').

The *isomorphic embedding* of T' into H' is defined for $k' \leq l'$. In this embedding node $t'(i, j)$ is assigned to PE $h'(i, j)$, for $0 \leq i \leq k' - 1$ and $0 \leq j \leq 2^i - 1$, and no nodes are assigned to PEs in H' at levels $\geq k'$.

The *shrink embedding* is defined for $k' > l'$. For levels 0 through $l' - 2$ the shrink embedding is same as the isomorphic embedding and level $l' - 1$ is handled as follows. Every leaf PE $h'(l' - 1, j)$ of H' is assigned the $2^{k'-l'+1} - 1$ nodes which represent the nodes in the subtree rooted at $t'(l' - 1, j)$ in T' . Observe that the isomorphic and the shrink embedding achieve $\delta = 1$ and $\lambda = 1$. However, the shrink embedding achieves a

very unbalanced load: one half of the PEs of H' are assigned 1 and the other half are assigned $2^{\frac{n'+1}{m'+1}} - 1$ nodes of T' .

We next describe the DLS (Different Level Subtree) embedding which embeds a $2m' + 1$ -node tree T' into an m' -PE tree H' . The DLS embedding achieves a dilation of unity, a congestion of $l' - 1$, and it distributes the leaves of T' evenly (i.e., $\mu_l = 2$). In order to achieve this, tree T' is partitioned at nodes on different levels and the subtrees rooted at these nodes are embedded using the isomorphic embedding, as follows. Let $T'_0, T'_1, \dots, T'_{l'-1}$ be l' subtrees of T' , where T'_i has height i and is rooted at node $t'(l' - i, 1)$, $0 \leq i \leq l' - 1$. Note that T' has height l' . We embed T'_i into H' using the isomorphic embedding and then assign all the nodes on the path from $t'(0, 0)$ to $t'(l', 0)$ to the root $h(0, 0)$. The DLS embedding assigns thus $l' + (l' - 1) = 2l' - 1$ interior nodes of T' to the root of H' and $l' - r - 1$ interior nodes of T' to every PE on level $r \geq 1$. The root is assigned 2 leaves of T' and every other PE of H' is assigned 1 leaf each. Figure 1 shows this embedding for $l' = 4$.

3 Embedding with Balanced Load

In this section we describe an embedding of T into H , referred to as Embedding 1, that achieves a balanced load, an optimal dilation δ of 1, and a congestion λ of $\min\{\lceil \frac{n}{m} \rceil, 2 \log n\}$. Let $k - 1$ and $l - 1$ be the heights of T and H , respectively, and let $c = 2^{k-l} = \frac{n+1}{m+1}$. Our embedding strategy used is recursive and encompasses two base cases. The first base case is applied when $k \leq l$. In this situation we simply embed T into H using the isomorphic embedding and achieve $\delta = \lambda = \mu = 1$. The second base case is applied when $l < k \leq l + \lceil \log l \rceil$ and the recursive strategy is used when $k > l + \lceil \log l \rceil$. The embeddings approaches used by the second base case and the recursive strategy have a common structure, even though they differ considerably in how to achieve the assignments. In both cases we obtain an initial embedding of T into H . A balancing step then reassigns nodes to obtain a balanced load for the second base case and a close to balanced load for the recursive strategy. In the latter, the balancing is completed by the recursion.

We next describe the second base case which handles the range $l < k \leq l + \lceil \log l \rceil$. Let T_r be the complete binary tree rooted at the root of T and having $2c - 1$ nodes. All

the nodes of T_r are assigned to $h(0,0)$, the root of H . Tree T_r contains c leaves which correspond to the c nodes on level $\log c$ in T and which have a total of $2c$ children. We partition these $2c$ nodes among $h(1,0)$ and $h(1,1)$, the two children of the root of H , as follows. PE $h(1,0)$ gets the c nodes that are the leftmost ones on level $\log c + 1$ in T and PE $h(1,1)$ gets the remaining nodes from level $\log c + 1$ (i.e., the c rightmost nodes). This assignment strategy continues. Let u be a node on level l_u in T that is assigned to PE $h(i,j)$, $l_u \geq \log c + 2$. If PE $h(i,j)$ is assigned at least $c/2$ other nodes of T that are to the right of node u on level l_u in T , then the two children of u are assigned to the left child of $h(i,j)$. Otherwise, they are assigned to the right child of $h(i,j)$. It is easy to see that the leaves of T are assigned to leaves of H . Once leaves have been assigned, the dilation is 1, the root $h(0,0)$ is assigned $2c - 1$ nodes, every other PE in H is assigned c nodes, and the congestion is $c = \frac{n+1}{m+1}$.

We next perform a balancing step that generates the final assignment in which every PE is assigned at most $c + 1 = \lceil \frac{n}{m} \rceil$ nodes. Let P_l (resp. P_r) be the path from $h(0,0)$ to $h(c/2 - 1, 0)$ (resp. $h(c/2 - 1, 2^{c/2-1} - 1)$). In the balancing step the number of nodes assigned to $h(0,0)$ decreases by $c - 2$ and the number of nodes assigned to every other PE on path P_l and P_r increases by one, respectively. PE $h(0,0)$ reassigns the leftmost $c/2 - 1$ leaves of T_r to $h(1,0)$ and the rightmost $c/2 - 1$ leaves of T_r to $h(1,1)$. After this step the dilation is still 1 and PEs on level 1 have $c + c/2 - 1 = 3c/2 - 1$ nodes assigned. These two PEs now each reassign $c/2 - 2$ of the nodes originally assigned to them to their child on path P_l or P_r , respectively. In general, a PE on level i and on path P_l or P_r receives $c/2 - i$ nodes from its parent and reassigns $c/2 - i - 1$ of the originally assigned nodes to its child on the corresponding path. If we select these $c/2 - i - 1$ nodes in an arbitrary way, the dilation would be 2. The following simple rule ensures dilation 1: a node u originally assigned to $h(i,j)$ can be reassigned to a child of $h(i,j)$ only when the parent of u was also reassigned. (This parent did get reassigned from the parent of $h(i,j)$ to $h(i,j)$.) It is straightforward to see that the balancing does not increase the congestion. The balancing procedure requires $c/2 < l$, which is satisfied when $k \leq l + \lceil \log l \rceil$.

We next describe the embedding strategy for $k > l + \lceil \log l \rceil$. It is not hard to see that a generalization of the strategy used in the second base case does not give the claimed

bounds. We now embed T into H in three steps, the third of which embeds a subtree of T into H recursively. The first step embeds T into a $2m + 1$ -node tree T' using the shrink embedding and then performs a DLS embedding of T' into H . Every leaf of T' represents a subtree of T of height $\log c - 1$ containing $c - 1$ nodes. Recall that the DLS embedding assigns $2l - 1$ interior and 2 leaf nodes of T' to the root $h(0, 0)$, and $l - i - 1$ interior and 1 leaf node to every other PE on level i . Hence, the root of H has $2l + 2c - 3$ nodes of T assigned and a PE on level i has $l + c - i - 2$ nodes assigned. The second step is a balancing step that alters the assignment so that the root of H has $2c - 1$ nodes and every other PE has c nodes assigned. The final recursive step embeds one of the subtrees of T of height $\log c - 1$ assigned to the root of H . We next describe the balancing step in detail.

As already stated, the balancing step reassigns nodes so that the root of H is assigned $2c - 1$ nodes and every other PE is assigned c nodes of T . Thus, PEs on level i , $0 \leq i \leq l - 3$, end up with fewer nodes, the number of nodes assigned to PEs on level $l - 2$ does not change, and the number of nodes assigned to every leaf PE on level $l - 1$ increases by 1. The maximum number of nodes that need to be reassigned for any PE is $2l - 2$. The first step of the embedding assigned to each PE a subtree of T containing $c - 1$ nodes. The root was assigned two such trees and only one of these trees will be involved in the balancing. The nodes reassigned will come from these subtrees. The balancing process needs $2l - 1 \leq c - 1$, which is satisfied when $k > l + \lceil \log l \rceil$. When $k \leq l + \lceil \log l \rceil$, we are not able to perform this type of balancing without increasing the dilation significantly. For this reason the second base case was introduced which handles this range of k by a different strategy.

The root $h(0, 0)$ has initially $2l + 2c - 3$ nodes assigned and $2l - 2$ of these nodes are reassigned. The root $h(0, 0)$ reassigns $l - 1$ nodes to its left child $h(1, 0)$ and $l - 1$ nodes to its right child $h(1, 1)$. After doing so, PE $h(1, j)$, $j = 0, 1$, has a total of $2l + c - 4$ nodes assigned to it, and it reassigns to its left and right child $l - 2$ nodes each. Assume we have reassigned nodes for PEs at levels $0, 1, \dots, i - 1$ in this fashion. Then a simple inductive argument shows that PE $h(i, j)$ on level i has $(l + c - i - 2) + (l - i) = 2l + c - 2i - 2$ nodes assigned to it. It reassigns $l - i - 1$ nodes to each of its children on level $i + 1$. This procedure ends when $i = l - 2$ at which point every PE on level $l - 2$ reassigns 1

node to each of its children on level $l - 1$.

We now describe which of the nodes are reassigned in the balancing step. Let $h(i, j)$ be a PE on level i , which has to reassign $l - i - 1$ nodes to each of its children. As already stated, the reassigned nodes come from the complete binary tree of height $\log c - 1$ assigned to $h(i, j)$ during the DLS embedding. Let T_α be this tree, $\alpha = \log c - 1$. PE $h(i, j)$ reassigns $l - i - 1$ nodes of the subtree rooted at $t_\alpha(1, 0)$ to its left child and $l - i - 1$ nodes of the subtree rooted at $t_\alpha(1, 1)$ to its right child. Here we only describe the reassignment process from $h(i, j)$ to its left child, since the reassignment process to the right child is symmetric. Let $l - i - 1 = b_s 2^s + b_{s-1} 2^{s-1} + \dots + b_0 2^0$, where b_r is either 0 or 1 and $s = \lfloor \log(l - i - 1) \rfloor$. For every index r such that $b_r = 1$, $0 \leq r \leq s$, let T_{r-1} be the subtree of height $r - 1$ rooted at $t_\alpha(\alpha - r + 1, 1)$. We then reassign $t_\alpha(\alpha - r, 0)$ and the nodes in T_{r-1} to PE $h(i + 1, 2j)$. Since T_{r-1} contains $2^r - 1$ nodes, a total of $l - i - 1$ nodes are reassigned to $h(i + 1, 2j)$.

Since the nodes reassigned to a PE v come from the subtree T_α initially assigned to the parent of v , the dilation is kept at 1. The congestion is increased by at most $\lceil \log(l - 1) \rceil$ and hence the congestion after the balancing step is at most $l - 1 + \lceil \log(l - 1) \rceil < 2(l - 1)$. This completes the description of the balancing step.

Finally, observe that after the balancing step, the root of T remains assigned to the root of H and $c - 1$ of the $2c - 1$ nodes assigned to PE $h(0, 0)$ form a complete binary tree of height $k - l - 1 = \log c - 1$. Let T^* be this tree. The last step recursively embeds T^* into H . The embedding of T^* achieves dilation 1, load $\lceil \frac{c-1}{m} \rceil$, and congestion at most $\min\{\lceil \frac{c-1}{m} \rceil, 2 \log n - 2 \log m\}$. Combining the embedding of T^* with the embedding of $T - T^*$, it is easy to see that the dilation remains 1 and the total number of nodes assigned to every PE of H is at most $c + \lceil \frac{c-1}{m} \rceil = \lceil \frac{n}{m} \rceil$. That the congestion is at most $2 \log n$ can be shown as follows. Let $\Lambda(k)$ be the congestion achieved when embedding a tree of height $k - 1$ into H (which has a fixed height of $l - 1$). Since the embedding of $T - T^*$ achieves a congestion of at most $2(l - 1)$, after combining the embeddings of $T - T^*$ and T^* we have

$$\Lambda(k) \leq 2(l - 1) + \Lambda(k - l) < 2 \log n.$$

This completes the embedding of T into H for the case when $k > l + \lceil \log l \rceil$. We now can state the following result.

Theorem 1 *An n -node complete binary tree T can be embedded into an m -PE complete binary tree H so that the dilation δ is 1, the load μ is $\lceil \frac{n}{m} \rceil$, and the congestion λ is $\min\{\lceil \frac{n}{m} \rceil, 2 \log n\}$.*

Proof: Follows from the discussion above. ■

We next briefly discuss the leaf and interior loads μ_l and μ_i , respectively, achieved by Embedding 1. Since Embedding 1 achieves a balanced load, we can restrict our attention to the leaf load μ_l . (The interior load μ_i is then $\lceil \frac{n}{m} \rceil - \mu_l$). For some values of k and l Embedding 1 achieves a nearly balanced leaf load, while for others it is rather unbalanced. In an embedding with a balanced leaf load every PE has at most $\lceil \frac{n+1}{2m} \rceil$ and at least $\lfloor \frac{n+1}{2m} \rfloor$ leaves of T assigned. When $l < k \leq l + \lceil \log l \rceil$, the second base case assigns no leaf nodes to some PEs of H and $\Theta(\frac{n+1}{m})$ leaf nodes to other PEs of H . For $l + \lceil \log l \rceil < k \leq 2l$, Embedding 1 consists of one recursive call and terminates with the first base case. It can be shown that the number of leaf nodes assigned to a PE is at most $\frac{n+1}{2(m+1)} + 2$ and at least $\frac{n+1}{2(m+1)} - \log m$. When $dl < k \leq (d+1)l$, for an integer $d \geq 2$, Embedding 1 achieves a fairly balanced leaf load. If $dl < k \leq dl + \lceil \log l \rceil$, then Embedding 1 consists of d iterations and it terminates using the second base case. The number of leaf nodes assigned to a PE of H can be shown to be at least

$$\sum_{i=1}^{d-1} (2^{k-il-1} - 2^{\lceil \frac{l}{2} \rceil}) < \lceil \frac{n+1}{4m} - \log n \rceil.$$

and at most

$$\sum_{i=1}^{d-1} (2^{k-il-1} + 1) + 2^{k-dl} < \lceil \frac{3(n+1)}{4m} + \frac{\log n}{\log m} \rceil.$$

Similar bounds hold for $dl + \lceil \log l \rceil < k \leq (d+1)l$.

We point out that it is possible to modify Embedding 1 so that it achieves a balanced l/i load with a dilation of 2 whenever k is a multiple of l . The details of this embedding are presented in [9]. Section 4 considers the problem of embedding T into H with a balanced l/i load in detail. It will be shown that any embedding strategy achieving a balanced l/i load and handling all values of k and l must have a dilation of at least 3.

We conclude this section with a brief discussion on another cost measure that is of interest in tree embeddings. A number of functions, when executed on a tree machine, require only communication between two levels of the tree at any time. Examples of such

functions are broadcasting and reduction functions (e.g., computing min and max). For this special flow of data the congestion is a too pessimistic bound and this motivates a new cost measure, the level-congestion. The *level-congestion* λ_l is the maximum congestion over any edge of H when only two adjacent levels of T are active at any time. It can be shown that the level-congestion achieved by Embedding 1 is $O(\min\{\frac{n}{m}, \frac{\log n}{\log m} + \log m\})$. However, one can do better. In [8, 9] we describe an embedding of an n -node tree T into an m -PE tree H that achieves a level-congestion of 4 and a dilation of 2. Load and congestion are as in Embedding 1. This improved embedding has a structure similar to that of Embedding 1, but it is more involved. The two main differences are in the way the second base case is handled (a completely different strategy is needed) and in the way nodes from the subtrees T_α are reassigned during the balancing step. We refer the interested reader to [8, 9] for details.

4 Embedding with Balanced 1/i Load

The main result of this section is an embedding of an n -node complete binary tree T into an m -PE complete binary tree H that achieves a balanced 1/i load. We refer to this embedding as Embedding 2. The dilation δ of Embedding 2 is $2\lceil\log\log m\rceil + 1$ and its congestion λ is $O(\log n)$. The general strategy is to first obtain an embedding of T into a $2m + 1$ -node tree T' using the shrink embedding, then embed T' into H efficiently, and then complete the embedding using a recursive step.

The heart of Embedding 2 is the embedding of T' into H with balanced 1/i load. Since T' contains $2m + 1$ nodes, this embedding has a leaf load of 2 and an interior load of 1. We obtain an initial, unbalanced embedding of T' into H by a DLS embedding and then refine the embedding in a delicate balancing step. While the balancing achieves a balanced 1/i load, it results in a dilation of $2\lceil\log\log m\rceil + 1$. We conjecture this dilation to be optimal. As mentioned in the previous section, for certain values of k and l there exists an embedding with a balanced 1/i load that achieves dilation 2. It appears that the “hardest” case is the one when the two trees differ in their heights by 1. For this situation we show that any embedding with a balanced 1/i load has to have dilation at least 3 when $l > 6$.

Section 4.1 describes how to embed an $2m+1$ -node tree T' into an m -PE tree H with a balanced l/i load. The embedding handling arbitrary values of k and l is described in Section 4.2. Section 4.3 contains the lower bound proof.

4.1 Embedding when $n = 2m + 1$

Let T' be $2m+1$ -node complete binary tree, $m = 2^l - 1$. We show how to embed T' into an m -PE tree H with a dilation of $2\lceil \log l \rceil + 1$, a balanced l/i load, and a congestion of $O(l)$. In this embedding exactly one PE of H , namely the root, is assigned 2 leaf nodes and 1 interior node of T' , and all the other PEs of H are assigned 1 leaf and 1 interior node each.

Assume without loss of generality that l is a power of 2. If l is not a power of 2, the bounds hold by using $\lceil \cdot \rceil$'s. The embedding of T' into H consists of 2 steps: a DLS embedding step and a balancing step. The DLS embedding assigns $2l-1$ interior and 2 leaf nodes of T' to the root of H , and $l-r-1$ interior and 1 leaf node of T' to every other PE of H at level r . Furthermore, it achieves unit dilation and a congestion of $l-1$. After the DLS embedding we already have a balanced leaf load. The balancing step described next reassigns only interior nodes of T' in order to achieve a balanced interior load. This reassignment causes the dilation to increase to $2\log l + 1$. The congestion remains $O(l)$.

The balancing step handles the left and right subtree of H independently. Let H' be one of these two subtrees containing $l-1$ levels (levels in H' are numbered $0, 1, \dots, l-2$). Within H' every interior node is pushed onto a higher level. If a node is assigned to PE v at level l_v in H' , then it will be pushed to a PE that is a descendent of v on level $l_v + r$, $r \leq \log l$. The balancing is done in a bottom-up fashion in $\log l - 1$ stages. The i^{th} stage considers subtrees H'' of height $2^i + i - 1$ rooted at PEs on level $l - 2^{i+1}$ in H' . At the beginning of the i^{th} stage the situation within each subtree H'' is as shown in Figure 2(a). The bottom i levels of H'' have no interior nodes assigned to them, and the top 2^i levels have the assignment of interior nodes as generated by the DLS embedding. During the i^{th} stage the interior nodes are pushed down so that at the end of the stage the top $i+1$ levels of H'' have no interior nodes assigned to them, and every PE on another level has exactly 1 interior node assigned to it. Furthermore, a node at a PE v in H'' is reassigned to a descendent of v at most $i+1$ levels below the level of v . Figure

2(b) shows the result of stage i on H'' . We next give an informal outline of the balancing step.

procedure BAL

Input: a tree H of height $l - 1$ and the assignment of interior nodes
generated by the DLS embedding.

Output: the embedding of T' into H .

1. for $p = 0$ to 1 do
 2. Let H' be the subtree of height $l - 2$ rooted at $h(1, p)$.
 3. for $i = 1$ to $\log l - 1$ do (* execute i^{th} stage *)
 4. for every PE v on level $l - 2^{i+1}$ in H' do
 Let H'' be the subtree of height $2^i + i - 1$ rooted at v ;
 5. RE_INT (H'', i).
 - endfor (* 1, 3 and 4 *)
 6. Reassign $2l - 2$ interior nodes assigned to $h(0, 0)$ to PEs on levels
 1 through $\log l$ so that every PE on these levels gets 1 interior node.
- end BAL.

The heart of the balancing step is the procedure RE_INT which does the actual reassignment of interior nodes. At the i^{th} stage procedure RE_INT works with subtrees H'' of height $2^i + i - 1$. Recall the assignment of interior nodes to PEs of H'' from Figure 2(a). A crucial observation is that the total number of interior nodes assigned to PEs at the top 2^i levels of H'' is same as the total number of PEs in the bottom $2^i - 1$ levels of H'' . Procedure RE_INT reassigns interior nodes to obtain the assignment of Figure 2(b). This is done in a bottom-up fashion by starting to reassign interior nodes to PEs at the bottom level of H'' and then marching upwards in H'' , level by level. We keep the dilation at $2(i + 1) + 1$ by doing the reassignments in subtrees S of height $i + 1$. Initially, the leaves of S correspond to the leaves of H'' . We reassign interior nodes assigned to the root and its left and right child of every such subtree S to the leaves of S .

The precise process of the reassignment in the r^{th} iteration is described next. Assume by induction hypothesis that we have reassigned interior nodes in $(r - 1)$ iterations so

that the assignment to PEs at level j in H'' with $j \leq 2^i - r - 1$ did not change (i.e., every PE has $2^{i+1} - j - 2$ interior nodes assigned to it), and every PE at level $2^i - r$ has $2^i - r$ interior nodes assigned to it. Every PE at level j with $2^i - r < j \leq 2^i + i - r$ has no interior node assigned to it, and every PE at level j with $j > 2^i + i - r$ has 1 interior node assigned to it. Observe that for $r = 1$ our hypothesis trivially holds. The r^{th} iteration considers subtrees S of H'' of height $i + 1$ rooted at PEs on level $2^i - r - 1$ in H'' . In every such subtree S , consider the subtree SL (resp. the subtree SR) of height i rooted at the left child (resp. right child) of $s(0, 0)$. There are 2^i leaves in SL. We first reassign all interior nodes assigned to $s(1, 0)$ to the leaves of SL. Since there are only $2^i - r$ interior nodes assigned to $s(1, 0)$ after $(r - 1)$ iterations, we reassign r interior nodes assigned to $s(0, 0)$ to the leaves in SL that did not get an interior node from $s(1, 0)$. A similar reassignment is done for the leaves of SR. Once the reassignment is done for all the subtrees in the r^{th} iteration, every PE at level $2^i - r$ in H'' has no interior node assigned to it, and every PE at level $2^i - r - 1$ has $(2^{i+1} - (2^i - r - 1) - 2) - 2r = 2^i - r - 1$ interior nodes assigned to it. The assignment to PEs at level j with $j \leq 2^i - r - 2$ has not changed. Since the leaves of S correspond to PEs at level $2^i + i - r$ in H'' , every PE at this level has 1 interior node assigned to it. Lemma 2 describes the situation after r iterations of RE_INT. There are $2^i - 1$ iterations in RE_INT and after the $(2^i - 1)^{\text{st}}$ iteration, every PE on level j in H'' with $j \leq i$ has no node assigned to it and every PE on level j in H'' with $j > i$ has 1 node assigned to it. This completes the description of procedure RE_INT. Thus the claimed final assignment of interior nodes in H'' follows.

Lemma 2 *Given is a tree H'' of height $2^i + i - 1$ and an assignment of interior nodes of dilation 1 in which:*

- (i) *every PE on level j has $2^{i+1} - j - 2$ nodes assigned for $0 \leq j < 2^i$, and*
- (ii) *every PE on level j has no node assigned for $2^i \leq j \leq 2^i + i - 1$.*

Then, after r iterations of procedure RE_INT the interior nodes are reassigned so that in H'' the dilation is at most $2(i + 1) + 1$ and

- (a) *every PE on level j has the same $2^{i+1} - j - 2$ nodes assigned as generated by the DLS embedding for $0 \leq j \leq 2^i - r - 2$,*

- (b) every PE on level $2^i - r - 1$ has $2^i - r - 1$ nodes assigned,
- (c) every PE on level j has no node assigned for $2^i - r \leq j \leq 2^i + i - r - 1$,
- (d) every PE on level j has 1 node assigned for $2^i + i - r \leq j \leq 2^i + i - 1$.

We now return to procedure BAL. It follows easily from RE_INT that after all $\log l - 1$ stages in BAL have been completed, the assignment of $2l - 1$ interior nodes to the root $h(0, 0)$ did not change, every PE on level j in H with $1 \leq j \leq \log l$ has no node assigned to it, and every PE on level j in H with $j > \log l$ has 1 node assigned to it. There are $2l - 2$ PEs at levels 1 through $\log l$ in H . In Step (6) of procedure BAL $2l - 2$ of the $2l - 1$ interior nodes at the root $h(0, 0)$ are reassigned arbitrarily to these PEs so that every one of them gets 1 interior node assigned. This completes the reassignment in the balancing step. We now can state the following result.

Lemma 3 *A $2m + 1$ -node complete binary tree T' can be embedded into an m -PE complete binary tree H so that the dilation is $2\lceil \log \log m \rceil + 1$, the leaf load is 2, the interior load is 1, and the congestion is $O(\log m)$.*

Proof: Recall that $m = 2^l - 1$. It is easy to see that the dilation achieved in the reassignment of interior nodes by the balancing step is $2\lceil \log l \rceil + 1$. In order to prove that the congestion is $O(l)$, consider the congestion over an edge (v_1, v_2) in H where v_1 is the parent of v_2 . Let i be the stage of procedure BAL so that at the end of stage i PEs v_1 and v_2 are reassigned 1 interior node each. Let H_q be the subtree of height q rooted at v_2 , where $q = i + 1$. The congestion of edge (v_1, v_2) is incurred by the nodes which are reassigned to PEs in the tree H_q and whose parent nodes are not reassigned to the PEs in H_q . There are $\sum_{j=0}^{q-1} 2^j = 2^q - 1$ PEs in H_q . Since every PE of H is reassigned 1 node, there are a total of $2^q - 1$ nodes which are reassigned to the PEs in H_q . It may happen that all of these nodes (i.e., nodes reassigned to the PEs in H_q) have their parent nodes not reassigned to the PEs in H_q . Hence, the total congestion of edge (v_1, v_2) is at most $O(2^q)$. Since q can be at most $\log l + 1$, the congestion is $O(l)$.

It is clear that the reassignment from $h(0, 0)$ in Step (6) results in a dilation of $2\lceil \log l \rceil + 1$ and a congestion of at most $O(\log l)$. Finally, consider the assignment of

leaf nodes done by the DLS embedding and the reassignment of their parent nodes by the balancing step. The balancing step reassigns interior nodes to at most distance $\lceil \log l \rceil$ from their original position in the DLS embedding. Since the assignment of leaf nodes does not change in the balancing step and since the dilation achieved in the DLS embedding is unity, the distance between a leaf node and its parent can be at most $\lceil \log l \rceil$. Thus, the dilation achieved in the embedding of T' into H is $2\lceil \log l \rceil + 1$ (which is caused by the balancing step). The congestion is increased by an additive factor of $\lceil \log l \rceil + 1$, but remains $O(l)$. Combining this with the bounds on dilation and congestion after the balancing step yields the claimed bounds and hence the lemma follows. ■

4.2 The General Embedding

We now describe how to use the result of Section 4.1 to obtain Embedding 2 which achieves a balanced $1/i$ load for all values of n and m . In the general case every PE of H is assigned at most $\lceil \frac{n+1}{2m} \rceil$ leaf nodes of T and at most $\lceil \frac{n-1}{2m} \rceil$ interior nodes of T . Embedding 2 achieves a dilation of $2\lceil \log \log m \rceil + 1$ and a congestion of $O(\log n)$. The embedding strategy is as follows.

If $k \leq l$, we use the isomorphic embedding to embed T into H . This achieves unit dilation, unit congestion, and unit load. If $k > l$, we first embed T , which is of height $k - 1$, into a $2m + 1$ -node tree T' of height l by performing the shrink embedding. We then obtain the final embedding in two steps: embedding T' into H and a recursive step.

In order to embed T' into H , we use the embedding procedure described in Section 4.1. This achieves a dilation of $2\lceil \log l \rceil + 1$ and a congestion of $O(l)$. Furthermore, every PE of H is assigned 1 interior node and 1 leaf of T' except the root, which is assigned 1 interior node and 2 leaves of T' . Since each leaf of T' contains $c - 1$ nodes of T (recall that $c = 2^{k-l}$), the root of H is assigned $2c - 1$ nodes of T and every other PE of H is assigned c nodes of T . Observe that, out of the $2c - 1$ nodes at the root, $c - 1$ nodes form a tree of height $k - l - 1$ in T . Let T^* be this tree. We recursively embed T^* into H . Let $T - T^*$ denote the tree after the nodes in T^* has been removed from T . Since the embedding of $T - T^*$ into H assigns $\frac{c}{2}$ leaf nodes of T to every PE of H and the embedding of T^* into H assigns at most $\lceil \frac{c}{2m} \rceil$ leaf nodes to every PE of H , the embedding of T into H assigns a total of at most $\frac{c}{2} + \lceil \frac{c}{2m} \rceil = \lceil \frac{n+1}{2m} \rceil$ leaf nodes to every PE of H . Similarly, the

embedding of T into H assigns at most $\lceil \frac{n-1}{2m} \rceil$ interior nodes to every PE of H . Let $\Lambda(k)$ (resp. $\Lambda(k-l)$) be the congestion of the embedding of T (resp. T^*) into H . Then

$$\Lambda(k) \leq O(l) + \Lambda(k-l) = O(k) = O(\log n).$$

This completes Embedding 2 and thus we have established the following result.

Theorem 4 *An n -node complete binary tree T can be embedded into an m -PE complete binary tree H with dilation δ of $2\lceil \log \log m \rceil + 1$ and congestion λ of $O(\log n)$ so that the leaf load μ_l is $\lceil \frac{n+1}{2m} \rceil$ and the interior load μ_i is $\lceil \frac{n-1}{2m} \rceil$.*

4.3 A Lower Bound

The previous sections showed that requiring a balanced l/i load seems to cause an increase in the dilation. In this section we show that when the trees differ in their height by one, every embedding with a balanced l/i load must have a dilation of at least 3. Note that when the trees differ in their height by one, a balanced l/i load implies every PE of H to be assigned exactly 1 leaf and 1 interior node of T' except one PE of H which is to be assigned 2 leaves and 1 interior node. Our lower bound proof holds for $l > 6$, where $m = 2^l - 1$. For $l \leq 5$, T' can be embedded into H with a dilation of 2. The lower bound is obtained by assuming that dilation 2 is possible and characterizing the assignments made to the leaf PEs of H . These characterizations establish relationships between sibling leaves and leaves in a common subtree of height 4 in H which lead to contradictions.

We first give some definitions and notations used throughout this section. We then give a simple argument showing that a dilation of 1 is not possible. We generalize the technique used to show that a dilation of 2 is not possible. For a PE v in H (excluding the one that has two leaf nodes assigned), let l and u be the leaf and interior node of T' assigned to it, respectively. Let $P(l, u)$ be the path between l and u in T' . When the path $P(l, u)$ contains 2 nodes that are on the same level in T' , we say that the path $P(l, u)$ is a *bent path*. If, in a bent path, the children of the interior node u are leaf nodes (i.e., u is on level $l-1$ in T'), we say that $P(l, u)$ is a *bpl* (*bent path with leaves*) *path*. See Figure 3(a) for an example of a bpl path. If $P(l, u)$ is not a bent path, we

say that it is a *straight path* (see Figure 3(b)). A leaf node l' adjacent to u is called a *leaf boundary node*, and an interior node u' adjacent to either l or u is called an *interior boundary node*. In Figure 3(a), nodes u_1 and u_2 are interior boundary nodes and leaves l_1 and l_2 are leaf boundary nodes. Any such boundary node has to be assigned to a PE at most distance δ from v in an embedding $\langle f, g \rangle$ with dilation δ .

We next give a simple argument showing that for $l \geq 3$ (i.e., H has height at least 2) a dilation of 1 cannot be achieved by an embedding that has a balanced l/i load.

Lemma 5 *A dilation of 1 is not possible in an embedding $\langle f, g \rangle$ with balanced l/i load when $l \geq 3$.*

Proof: Let l and u be the two nodes assigned to a leaf PE v . Assume, without loss of generality, that both the parent and the sibling of v have 1 leaf node assigned to them. If $P(l, u)$, the path from l to u , has length 2 or more, then it has at least 2 interior boundary nodes which need to be assigned to the parent of v . Since this is not possible, $P(l, u)$ must have length 1. Let l_s and u_s be the two nodes assigned to v_s , the sibling of v . $P(l_s, u_s)$ must also be a path of length 1. $P(l, u)$ and $P(l_s, u_s)$ together have 2 leaf boundary nodes which need to be assigned to the common parent of v and v_s . This is not possible and the lemma follows. ■

For the remainder of this section, let $\langle f, g \rangle$ be an embedding of T' into H with a dilation of 2 and a balanced l/i load. Assume $l > 6$. Let H' be a subtree of H of height 4 such that the leaves of H' correspond to leaves in H , no PE in H' has two leaf nodes of T' assigned to it, and no leaf PE of H' has the root of T' assigned to it. Since the height of H is at least 6, there always exists a subtree H' satisfying the above conditions. Let v be a leaf PE in H' having an interior node u and a leaf node l assigned to it. The next two lemmas partially characterize $P(l, u)$, the path from l to u in T' .

Lemma 6 *If the path $P(l, u)$ is a bent path, then it is a bpl path.*

Proof: Assume that $P(l, u)$ is bent path, but not a bpl path. Then, $P(l, u)$ has a total of 4 interior boundary nodes which need to be assigned to either the sibling, the parent, or the grand-parent of v . This is not possible, since each PE is assigned only one interior node. ■

Lemma 7 *If the path $P(l, u)$ is a straight path, then $P(l, u)$ has length at most 2.*

Proof: If $P(l, u)$ is a straight path having length at least 3, then $P(l, u)$ has 4 interior boundary nodes (recall that u cannot be the root of T'). Hence, the lemma follows. ■

We say that PE v is of type 1, 2, or b , when the path $P(l, u)$ is a straight path of length 1, length 2, or a bent path with leaves, respectively. Consider two consecutive leaves v_0 and v_1 in H' belonging to a common subtree of height 1.

Lemma 8 *PEs v_0 and v_1 cannot both be of type 2.*

Proof: If v_0 and v_1 are both of type 2, they together have at least 5 interior boundary nodes. However, only two interior boundary nodes can be accommodated in an embedding of dilation 2. ■

Consider now four consecutive leaves v_0, v_1, v_2 , and v_3 in H' belonging to a common subtree of height 2. Let H'_2 be this subtree and r_2 be its root. The next lemma shows that it is not possible for every v_i to be of type 1 or b .

Lemma 9 *At least one v_i must be of type 2, $0 \leq i \leq 3$.*

Proof: If every v_i is of type 1 or b , four interior nodes on level $l - 1$ in T' are assigned to the leaves of H'_2 . Hence, eight leaves of T' need to be assigned at distance at most 2 from the leaves of H'_2 . There only exist seven positions for these eight leaves and thus the lemma follows. ■

From Lemma 8 we know that two siblings leaves in H' cannot both be of type 2. The remaining possible assignments to the leaves of H'_2 are either exactly one of the v_i 's is of type 2, or one of v_0 or v_1 and one of v_2 or v_3 are of type 2. We do not get a contradiction by considering subtree H'_2 alone. We show that these type assignments make certain requirements on which nodes need to be placed at the parent (or grand-parent) of r_2 , the root of H'_2 . Combining these requirements with those of other subtrees of H' leads to the final contradiction.

Lemma 10 *If one of v_0 or v_1 is of type 2 and one of v_2 or v_3 is of type 2, then the leaf node to be assigned to r_2 and the leaf node to be assigned to the parent of r_2 is determined by the assignments to the v_i 's.*

Proof: W.l.o.g. assume that v_0 and v_2 are of type 2. Let u_i and l_i be the interior and leaf node assigned to PE v_i , respectively, $0 \leq i \leq 3$. If u_0 and u_2 do not come from a common subtree of height 3 in T , they together have a total of six interior boundary nodes. These interior boundary nodes need to be assigned to PEs in H'_2 , which is not possible. Hence, let T_3 be the subtree of T' containing u_0 and u_2 . Figure 4 shows one such possible assignment of nodes of T_3 to the PEs of H'_2 (other assignments are symmetric). Let node a be the root of T_3 . In order to achieve a dilation of 2, node a needs to be assigned to r_2 , the root of H'_2 . This holds since r_2 is the only PE which is at a distance of at most 2 from v_0 and v_2 . All the other interior nodes of T_3 also need to be assigned to PEs in H'_2 . Thus, one of the eight leaf nodes in T_3 is required to be assigned to r_2 and another of these leaf nodes is required to be assigned to the parent of r_2 . ■

Consider now the situation when exactly one of the v_i 's is of type 2. Without loss of generality, let it be v_0 .

Lemma 11 *If v_0 is of type 2 and the other v_i 's are of type 1 or b, then*

1. *the leaf to be assigned to r_2 , and*
2. *one of the leaves assigned to either the parent of r_2 , the sibling of r_2 or the grand-parent of r_2*

is determined by the assignments to the v_i 's.

Proof: Let T_2 be the subtree of T' of height 2 containing l_0 and u_0 . Neither of the interior nodes u_2 and u_3 can be in T_2 (this would cause a dilation of 4). If both l_2 and l_3 are in T_2 , then a total of eight leaf nodes needs to be assigned to seven possible locations in H'_2 (since v_2 and v_3 must now both be of type b), which is not possible.

Assume that either one of l_2 or l_3 is in T_2 or none of the nodes assigned to v_2 and v_3 are in T_2 . Interior nodes u_2 and u_3 together have four leaf boundary nodes that need to be placed at distance at most 2 from PEs v_2 and v_3 . This implies that a leaf node from T_2 has to be assigned to r_2 . Interior node u_0 has four leaf nodes at distance 2 which need to be assigned to PEs at most distance 4 from v_0 . H'_2 can accommodate at most seven leaf nodes. Thus, one of the leaves from T_2 is required to be assigned to either the parent of r_2 , the sibling of r_2 , or the grand-parent of r_2 . ■

Let H'_0, H'_1, H'_2 , and H'_3 be the four subtrees of height 2 of H' whose leaves correspond to leaves of H' . Let r' be the root of H' and r_0 and r_1 be the left and right child of PE r' , respectively. Subtree H'_i is of type $s2$ if exactly one of its leaves is of type 2 and it is of type $d2$ if two of its leaves are of type 2, $0 \leq i \leq 3$. It is not possible for both the H'_0 and H'_1 to be of type $d2$. When H'_0 is of type $d2$, a leaf is required to be assigned to r_0 and it is easy to see that H'_1 requires a different leaf to be assigned to PE r_0 . A similar statement holds for H'_2 and H'_3 .

Thus, w.l.o.g., assume that H'_0 is of type $s2$ and H'_1 is either of type $d2$ or $s2$. Combining the leaf requirements for H'_0 and H'_1 requires a leaf to be assigned to r' , the root of H' . This holds since, from Lemmas 10 and 11 it follows that the roots of H'_0 and H'_1 have already a leaf node assigned and two more leaf nodes need to be assigned to PEs r_0 and r . A similar statement holds for H'_2 and H'_3 . Thus, PE r' needs to accommodate two leaf nodes of T' in order to achieve a dilation of 2. This is not possible and we thus have proven the following.

Theorem 12 *An embedding of a $2m + 1$ -node tree T' into an m -PE tree H with a balanced $1/i$ load has a dilation of at least 3 when $m > 2^6 - 1$.*

When the host tree H has height at most 4, T' can be embedded with a dilation of 2 and a balanced $1/i$ load. When H has height 4, the embedding can easily be obtained by having H correspond to the tree H' used in the lower bound proof and using the type assignments that assign two leaves of T' to the root of H' .

5 Weighted Embeddings

In this section we consider the problem of embedding a tree T into a tree H when every edge of T has a nonnegative weight associated with it and these weights are to be taken into account by the congestion. More formally, let $\langle f, g \rangle$ be an embedding of T into H . If an edge e of T has weight w_e , then the path $g(e)$ in H has weight w_e . The weighted congestion λ_w is defined as the maximum over the sum of weights of all the paths of T passing through an edge in H .

The weights on the edges of T may represent, for example, communication costs over the edges. Unfortunately, but not surprisingly, we show that for arbitrary weights

the problem of embedding T into H with a minimum weighted congestion is NP-hard. However, a number of applications do not have arbitrary weights, but weights of a special structure associated with the edges. One such special structure are exponentially decreasing weights. The edges of T have *exponentially decreasing weights* when every edge incident to a node at level i and a node at level $i + 1$ has a weight of $w/2^i$, for some $w = 2^r > 0$ and $0 \leq i \leq k - 2$. Exponentially decreasing weights arise, for example, when T models the amount of data to be transferred between processes in a divide-and-conquer strategy. Also, exponentially decreasing weights are obtained when every mesh in a Tree-of-Meshes network [13] is abstracted to a single node and the weights on the edges of the resulting binary tree reflect the number of original communication links. For the case of exponentially decreasing weights, our Embedding 1 achieves a dilation δ of 1, a weighted congestion λ_w of at most $3w$, and a balanced load. We show that the achieved weighted congestion is asymptotically optimal by proving a lower bound of $\lambda_w = \Omega(w)$.

We first present the NP-hardness result for the case of arbitrary weights. We refer to the decision version of the problem as the *minimum weighted congestion problem*. We are given an n -node complete binary tree T in which every edge e of T has a weight of $w_e \geq 0$, an m -PE complete binary tree H , and a nonnegative integer W . We are supposed to decide whether there exists an embedding of T into H with balanced load and $\lambda_w \leq W$.

Theorem 13 *The minimum weighted congestion problem is NP-complete.*

Proof: Given an embedding $\langle f, g \rangle$ of T into H , it is easy to check in time polynomial in n and m whether $\langle f, g \rangle$ achieves balanced load and has a weighted congestion of at most W . Thus the minimum weighted congestion problem is in NP. We show the NP-completeness by reducing the *partition problem* to this one. Consider an instance of the partition problem, where $S = \{a_1, a_2, \dots, a_{n'}\}$. Assume, without loss of generality, that $n' = 2^{k'}$ for some positive even integer k' . Let $n = 4n' - 1$ and let $T = (V_T, E_T)$ be an n -node complete binary tree having height $k - 1$, and H be a 3-PE complete binary tree of height 1. Since $k = k' + 2$ is an even integer, it can be easily shown that $\mu = \lceil n/3 \rceil = 2^0 + 2^2 + \dots + 2^{k-4} + 2^{k-2}$. Let T_{2i-1} be the subtree of T rooted at node

$t(k/2 - i, 1)$ and having height $2i - 1$, $1 \leq i \leq k/2 - 1$. Let $R = (V_R, E_R)$ be the subtree of T that consists of the nodes in the subtree T_{2i-1} and the parent of the root of T_{2i-1} for $1 \leq i \leq k/2 - 1$, and the node $t(\frac{k}{2} - 1, 0)$. In Figure 5 we show the subtree R when $k = 10$. It is easy to see that the number of nodes in R is μ . The weights on the edges of T are set as follows.

Let $M = \sum_{i=1}^{n'} a_i$. Every edge in R has weight $M + 1$. Let e_i be the i^{th} edge, counting from the left in the canonical drawing of a tree, between a leaf and its parent node in the right subtree of T , $1 \leq i \leq n'$. Then edge e_i has weight a_i , $1 \leq i \leq n'$. All the remaining edges of T have weight 0. We now show that there is a solution to the partition problem on the set S if and only if the embedding of T into H with a balanced load has a weighted congestion of $M/2$.

Assume that there exists a partition of S such that $\sum_{a_i \in S'} a_i = \sum_{a_i \in S-S'} a_i = \frac{M}{2}$. It is straight forward to see that an embedding of T into H achieving $\lambda_w = M/2$ and balanced load can be constructed. Assume now that there exists an embedding $\langle f, g \rangle$ with a balanced load and a weighted congestion $\lambda_w = M/2$. We first show that all the nodes in R must have been assigned to the root of H . Since every edge in R has a weight of $M + 1$, an edge $e \in E_R$ cannot be mapped to an edge of H . Hence, all the nodes in R are assigned to a PE v of H . Since, $|R| = \mu$, PE v has μ nodes assigned to it. If PE v is a leaf PE, then all the edges which have weight a_i must be mapped to the edge between the leaf PE v and its parent. Since this yields a weighted congestion of M , PE v must be the root of H .

The nodes of T in $V_T - V_R$ are now divided evenly between the left and right child of the root of H . The edges that contribute to the congestion are the edges incident to the leaf nodes in the right subtree of T . These leaf nodes have their parent already assigned to the root of H . In order to achieve a congestion of $M/2$, these leaf nodes are partitioned into two sets, each yielding a congestion of at most $M/2$. The remaining edges incident to nodes that are assigned to the leaves of H do not contribute to the congestion. Thus, if there exists an embedding with $\lambda_w = M/2$, the partition of the leaf nodes in the right subtree of T represents a solution to the partition problem on set S . This concludes the proof of the NP-completeness and hence Theorem 13 follows. ■

We now consider the case when the weights on the edges of T are exponentially decreasing. Recall that now an edge incident to nodes at level i and $i + 1$ in T has a weight of $w/2^i$, for $0 \leq i \leq k - 2$. As stated earlier, Embedding 1 described in Section 3 achieves $\delta = 1$, $\lambda_w \leq 3w$ for exponentially decreasing weights. A natural question is whether there exists an embedding of T into H that achieves a smaller congestion for certain values of n and m . We answer this question by proving that $\lambda_w = \Omega(w)$, for any embedding achieving a balanced load. This lower bound shows that Embedding 1 is optimal within a constant factor.

Theorem 14 *Given is an n -node complete binary tree T with exponentially decreasing weights on the edges of T and an m -PE complete binary tree H with $n \geq m > 16$. Then any embedding $\langle f, g \rangle$ of T into H achieving a balanced load must have a weighted congestion at least $w/2$, i.e., $\lambda_w = \Omega(w)$.*

Proof: Intuitively, the idea is the following: Let v be a PE of H and assume that the sum of the weighted congestion on all the edges incident to v in H is W . Since there are at most 3 incident edges, the embedding has a weighted congestion of at least $W/3$. Hence, a lower bound on W gives a lower bound on λ_w . We will show that in every embedding there exists a PE v such that W is at least $3w/2$ and thus $\lambda_w = \Omega(w)$.

Throughout this proof we choose v to be the PE that has $t(0,0)$ (the root of T) assigned. As before, let $n = 2^k - 1$ and $m = 2^l - 1$. In order to prove the lower bound on W we consider two cases depending on whether or not leaf nodes of T are assigned to v .

Case 1 : *No leaf node of T is assigned to PE v .*

In this case we show that W is at least $2w$, which is greater than $3w/2$. Let R be the subtree of T assigned to v that includes the root $t(0,0)$ of T . Let $R^{i,j}$ be the subtree of R that contains node $t(i,j)$ at level i in T and all the descendants of $t(i,j)$ which are in R . We first show by induction that every such subtree $R^{i,j}$ of R contributes $w/2^{i-1}$ to W . Let $h \leq k - 2$ be the highest level such that node $t(h,j)$, for some j , is in R . Since no leaf node of T is in R , the left and right child of $t(h,j)$ are not assigned to v . The edges to the children of $t(h,j)$ have weight $w/2^h$ each, and hence the subtree $R^{h,j}$ that contains only node $t(h,j)$, contributes $2 * w/2^h = w/2^{h-1}$ to W .

Assume by induction hypothesis that every subtree $R^{i+1,j'}$ of R contributes $w/2^i$ to

W . Recall that $R^{i+1,j'}$ contains a node $t(i+1, j')$ and all the descendants of $t(i+1, j')$ that are in R . Consider a node $t(i, j)$ at level i of T which is in R , and consider the subtree $R^{i,j}$. We show that the subtree $R^{i,j}$ contributes $w/2^{i-1}$ to W . If one of the children of $t(i, j)$ is not assigned to v , then the edge to this child has weight $w/2^i$. If one of the children of $t(i, j)$ is assigned to v , then it is in R and thus by induction hypothesis the corresponding subtree $R^{i+1,j'}$ (with $j' = 2j$ or $2j + 1$) contributes $w/2^i$ to W . Now, it is easy to see in all the possible combinations that the subtree $R^{i,j}$ contributes $w/2^i + w/2^i = w/2^{i-1}$ to W .

Since we assumed that $t(0,0)$ is assigned to v , the subtree $R^{0,0}$, which equals R , contributes $w/2^{0-1} = 2w$ to W . Hence, in Case 1 W is at least $2w$.

Case 2 : *Leaf nodes of T are assigned to PE v .*

In order to prove the lower bound on W we now augment the tree T by one level. Let T' be this tree, i.e., the one in which 2^k nodes have been added to form a complete binary tree of height k . Tree T' also has exponential decreasing weights and an edge between a leaf node of T' and its parent has weight $w/2^{k-1}$. We modify the embedding $\langle f, g \rangle$ to obtain an embedding $\langle f', g' \rangle$ of T' into H as follows. The assignment of the nodes of T' on the first k levels is same as the one for the nodes of T in $\langle f, g \rangle$. The leaf nodes of T' are assigned as follows. Let v be the PE that has root node of T' assigned to it. For every node u on level $k-1$ of T' that is assigned to v , we assign the two children of u to one of the PEs adjacent to v in H . For every other PE v' of H , we assign the children of a node on level $k-1$ of T' that is assigned to v' to PE v' itself. In this new embedding every PE of H has at most 5μ and at least $\mu-1$ nodes of T assigned to it. The upper bound is achieved when PE v is a leaf PE and the maximum number of leaf nodes of T are assigned to v and to its parent v_0 in the embedding $\langle f, g \rangle$. In such a case PE v_0 has 5μ nodes of T' assigned to it in the embedding $\langle f', g' \rangle$.

Recall that v is the PE that has the root node of T' assigned to it. The subtree R of T' assigned to v that includes $t'(0,0)$ does not contain leaf nodes of T' . The proof of the lower bound for Case 1 does not make use of the assumption that the embedding has a balanced load and it thus holds when at most 5μ and at least $\mu-1$ nodes are assigned to a PE. We apply Case 1 to embedding $\langle f', g' \rangle$ and it follows that the sum W' of congestion on all the edges incident to v in H is at least $2w$. PE v is assigned at most μ

nodes in $\langle f, g \rangle$. Hence, at most 2μ edges incident to the leaf nodes of T' contribute to W' . Every such edge has weight $w/2^{k-1}$ and thus $W > W' - 2\mu w/2^{k-1} \geq 2w - 2\mu w/2^{k-1}$ in the embedding $\langle f, g \rangle$. Since $n \geq m > 16$, we have $k \geq l \geq 5$ and thus

$$\begin{aligned} W &> 2w - 2\mu \frac{w}{2^{k-1}} = 2w - \frac{2w}{2^{k-1}} \lceil \frac{n}{m} \rceil > 2w - \frac{2w}{2^{k-1}} \left(\frac{n}{m} + 1 \right) \\ &= 2w - \frac{4w}{2^k} \left(\frac{2^k - 1}{2^l - 1} + 1 \right) > 2w - \frac{4w}{2^l - 1} - \frac{4w}{2^k} > \frac{3w}{2}. \end{aligned}$$

We thus conclude that in both Case 1 and Case 2, the sum of the congestion on all the edges connecting v is at least $3w/2$ in any embedding $\langle f, g \rangle$. Since v has degree at most 3, $\lambda_w \geq W/3 \geq w/2$ and hence Theorem 14 follows. ■

We conclude this section by stating that Theorems 13 and 14 hold for the following type of unbalanced embeddings. We say that an embedding of load μ achieves an α -load, if every PE of H has at least $\alpha\mu$ nodes of T assigned to it, $0 \leq \alpha \leq 1$. Recall that by definition, every PE has at most μ nodes assigned to it. It is straightforward to see that minimum weighted congestion problem remains NP-complete for embeddings with an α -load, where α is a positive constant. We refer the reader to [9] for the necessary changes in the lower bound proof.

6 Conclusions

In this paper we presented two embeddings of an n -processor complete binary tree architecture T into an m -processor complete binary tree architecture H , $n > m$. One embedding achieves constant dilation and a balanced load with an unbalanced l/i load. The other embedding achieves a nonconstant dilation and a balanced l/i load. We also showed that any embedding achieving a balanced l/i load for all values of n and m must achieve a dilation of at least 3. We conjecture that our second embedding achieves a dilation which is optimal within a constant factor; i.e., dilation must be $\Omega(\log \log m)$. We consider it unlikely that the techniques used in the lower bound proof of Section 4.3 generalize so that the gap between 3 and $O(\log \log m)$ can be closed. The main reason appears to be the inability to easily classify the paths $P(l, u)$ and the resulting exponential growth in the number of cases to be considered. For the case when the edges

of T have weights associated we proved the general embedding problem to be NP-hard. We considered exponentially decreasing weights and we proved a lower bound on the weighted congestion of $\Omega(w)$ for any embedding with a balanced load.

References

- [1] S. G. Akl. *The Design and Analysis of Parallel Algorithms*. Prentice Hall, 1989.
- [2] R. Aleliunas and A. Rosenberg. On embedding rectangular grids into square grids. *IEEE Trans. on Computers*, V-31:907–913, 1982.
- [3] J. L. Bentley and H. T. Kung. A Tree Machine for Searching Problems. In *the Proceedings of the International Conference on Parallel Processing*, pages 257–266, 1979.
- [4] S. Bhatt, F. Chung, F. T. Leighton, and A. Rosenberg. Optimal simulations of tree machines. In *the Proceedings of 27th Symposium on Foundation of Computer Science*, pages 274–282, 1986.
- [5] S. A. Browning. *The Tree Machine: A Highly Concurrent Computing Environment*. PhD thesis, Computer Science Dept., California Institute of Technology, 1980.
- [6] F. Berman and L. Snyder. On Mapping Parallel Algorithms into Parallel Architectures. *JPDC*, 4:439–458, 1987.
- [7] J. P. Fishburn and R. A. Finkel. Quotient networks. *IEEE Transactions on Computers*, C-31(4):288–295, April 1982.
- [8] A. K. Gupta and S. E. Hambrusch. Embedding Large Tree Machines into Small Ones. In *the Proceedings of the Fifth MIT Conference on Advanced Research in VLSI*, pages 179–198, March 1988.
- [9] A. K. Gupta. On the Relationship between Parallel Computation and Graph Embeddings. Ph.D. Thesis, Purdue University. August 1989.
- [10] J. -W. Hong, K. Mehlhorn, and A. Rosenberg. Cost trade-offs in graph embeddings, with applications. *JACM*, 709–728, 1983.
- [11] S. R. Kosaraju and M. Atallah. Optimal simulations between mesh connected array of processors. *Journal of the Association of Computing Machinery*, 35(3):635–650, July 1988.
- [12] R. Koch, F. T. Leighton, B. Maggs, S. Rao, and A. Rosenberg. Work-preserving emulations of fixed-connection networks. In *the Proceedings of 21th STOC*, pages 227–240, 1989.

- [13] F. T. Leighton. *Complexity Issues in VLSI*. MIT press, 1983.
- [14] G. A. Mago. A Network of Microprocessors to Execute Reduction Languages, Parts 1 and 2. *International Journal of Computer and Information Sciences*, pages 349-385 and pages 435-471, October 1979 and December 1979.
- [15] A. L. Rosenberg. Preserving proximity in arrays. *SIAM Journal on Computing*, 443-460, 1979.
- [16] A. L. Rosenberg. On Validating Parallel Architectures via Graph Embeddings. *Concurrent Computations*, S.K. Tewksbury, Editor. Plenum Press, 99-115, 1988.
- [17] F. C. Sang and I. H. Sudborough. Embedding Large Meshes into Small Ones. To appear in the *Proceedings of the IEEE Symposium on Circuits and Systems*, 1990.

(a) Tree T used in the DLS embedding.

(b) Tree H' in the DLS embedding showing assignment of T when $l=4$.

Figure 1: The DLS embedding.

Figure 2: Schematic Process in the i^{th} stage of procedure BAL.

(a) A bpl path

(b) A straight path
of length 3

Figure 3: Subtrees of T' with paths $P(l, u)$ shown in bold.
(u_i and l_j are boundary nodes.)

(a) Subtree T_3

(b) Subtree H'_2

Figure 4: An assignment of Lemma 10.

Figure 5: The subtree R of Theorem 13 for $k = 10$.