Computational Properties of Mesh Connected Trees: Versatile Architectures for Parallel Computation

Kemal Efe and Antonio Fernández*
The Center for Advanced Computer Studies,
University of Southwestern Louisiana, Lafayette, LA 70504

Abstract

Recently, the mesh connected trees *(MCT) network has been proposed as a possible architecture for parallel computers. MCT networks are obtained by combining complete binary trees using the cross product operation. This paper focuses on structural, embedding, routing, and layout properties of the MCT networks. We show that MCT networks are computationally more powerful than grids and complete binary trees, and at least as powerful as meshes of trees (MOT). Analysis of VLSI complexity shows that the additional power is obtained without asymptotically increasing the layout area with respect to the grid of at least 3 dimensions or to the MOT of any number of dimensions. A variation of the basic architecture with same maximum vertex degree and same asymptotic area complexity is also investigated. This variation contains the torus as a subgraph as well as the MOT, further increasing the computational power of the basic architecture. These results suggest that the basic MCT network and its variant are suitable architectures for a large class of massively parallel computations.*

Introduction

The Mesh Connected Trees (MCT) network is the multidimensional cross product of complete binary trees. It was originally presented as a parallel architecture in [2]. The major advantage offered by the MCT architecture is that it can perform grid computations with a small constant slowdown and Mesh of Trees (MOT) computations without any slowdown. While grids and meshes of trees are powerful architectures for certain computation patterns, neither one of these architectures is very effective in the application domain of the other. MCT architectures and their variants bridge this gap between grids and meshes of trees since they are able to emulate both networks.

We use the notation MCT to refer generically to the class of mesh connected trees networks, while we use $MCT_r(N)$ when we refer specifically to the N^r-node r-dimensional mesh connected trees obtained as the cross product of N-node complete binary trees. Figure 1.(a) shows the 49-node 2-dimensional mesh connected trees, $MCT_2(7)$.

*This author's research has been supported by the Spanish Ministry of Education and Science, in collaboration with the Fulbright Program.

Recently, several new interconnection networks based on the product operation have been proposed. Ganesan and Pradhan [4] introduced the product graph obtained by combining hypercubes and de Bruijn networks. Rosenberg [8], introduced the product of de Bruijn graphs as a potential parallel architecture and analyzed several of its computational properties. Youssef [10] defined new product networks by combining the hypercube with various other networks and presented general results for product networks in terms of the properties of the factor networks.

Efe and Fernández [2] restricted their study to product networks whose factor graphs are all isomorphic. Structural and embedding properties were studied for these networks and three specific cases were analyzed in some detail. The MCT graph was among the cases studied, and in our knowledge the MCT network first appeared there as a potential candidate for a parallel architecture.

An interesting property of the MCT is that the size of the network can be increased either by increasing the size of the base tree (N) or by increasing the number of dimensions (r). While r is likely to exceed one, for practical purposes it can be considered as constant (and so shall be considered in this paper), and the network grows by growing the size of the binary trees. This facility for modular growth without increasing the number of dimensions, and therefore the vertex degree, implies that a parallel computer based on the MCT topology can be expanded in size without increasing the number of I/O channels at the processors.

An interesting variant of the MCT network, which we call MCXT, is obtained as the cross product of an extended form of complete binary trees as follows: we first connect the leaves of the binary tree by a straight line to obtain an extended tree (XT), and then construct the product of the resultant XT graph. While the added connections do not increase the asymptotic VLSI complexity, the computational properties are improved significantly.

This paper is organized as follows: in the next section we present definitions and terms used throughout the paper. The subsequent sections address, respectively, the structural properties of MCT, several embedding capabilities, routing properties, and VLSI layout area. Subsequently we present a simple extension to the basic network. Finally, conclusions and open problems are presented.

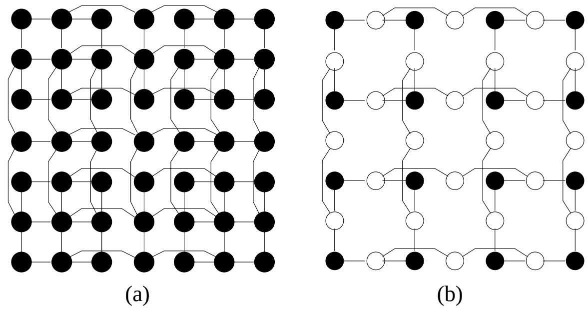

Figure 1: (a) 49-node 2-dimensional mesh connected trees $MCT_2(7)$ and (b) two dimensional mesh of 4-leaf trees (leaves are shown as dark nodes).

Definitions

In this paper a interconnection network is seen as an undirected graph, whose vertices represent processors and whose edges represent bidirectional links between the processors. We use the terms graph and network interchangeably. Since the MCT is obtained from the complete binary tree, we start by formally defining this latter graph.

The **h-level complete binary tree**, $T(h)$, is the graph whose vertices comprise the set $\{1...N\}$, where $N = 2^h - 1$, and whose edges connect each vertex $u < 2^{h-1}$ with the vertices $2u$ and $2u+1$. The vertices at level j are labeled from 2^{j-1} to $2^j - 1$, for $j = 1...h$. The vertex labeled 1, al level 1, is the root of the tree. The vertices at level h are the leaves of the tree.

The **N^r-node r-dimensional mesh connected tree**, $MCT_r(N)$, is the graph whose vertices comprise all the r-tuples $x = x_{r-1}...x_1x_0$, such that, for $i = 0...r-1$, every x_i is a vertex of $T(h)$, where $N = 2^h-1$, and the pair (x, y) defines an edge in $MCT_r(N)$ if and only if x and y differ in exactly one index position i and (x_i, y_i) is an edge in $T(h)$. A node $x = x_{r-1}...x_1x_0$ where x_i is a leaf of $T(h)$, for $i = 0...r-1$, is a leaf of $MCT_r(N)$. The node $x = x_{r-1}...x_1x_0$ where $x_i = 1$ for $i = 0...r-1$ is the root of $MCT_r(N)$.

The **N^r-node r-dimensional grid** (resp. **torus**) is the graph whose vertices comprise all the r-tuples $x = x_{r-1}...x_1x_0$, such that $x_i \in \{0...N-1\}$, $i = 0...r-1$, and whose edges connect any pair of nodes x and y if and only if x and y differ in exactly one index position i and $|x_i - y_i| = 1$ (resp. $x_i = (y_i + 1) \ mod \ N$ or $y_i = (x_i + 1) mod N$).

The **r-dimensional mesh of N-leaf trees** is the graph obtained from the N^r-node r-dimensional grid by substituting the linear connections along each dimension by N-leaf complete binary trees. The leaves of the trees are the original nodes of the grid, and additional nodes are introduced to obtain the internal nodes of the trees (see Figure 1.(b)).

An embedding of a "guest" graph G in a "host" graph H is a one-to-one mapping of the vertices of G into the vertices of H and of the edges of G into paths in H. The main cost measures of an embedding are the *dilation*, the length of the longest path to which an edge of G is mapped, and the *congestion*, the maximum number of paths (images of edges of G) sharing any edge of H.

Structural Properties

Several structural properties of the MCT where presented in [2]. There it was shown that $MCT_r(N)$ has N^r vertices and $rN^{r-1}(N-1)$ edges. The minimum vertex degree is r and the maximum vertex degree is $3r$. This makes the vertex degree bounded if the number of dimensions is bounded. Since grids and meshes of trees with more than 3 dimensions are rarely used, we expect that r will be generally small for MCT also, and the vertex degree wil be also small and constant for practical purposes.

It was also shown in [2] that $MCT_r(N)$ has diameter $2r(h-1)$ (logarithmic in the number of nodes) and bisection width of at least $\frac{N^{2r}-1}{N^{r-1}(N^2-1)}$. These attributes of the MCT, coupled with its small vertex degree make it an attractive architecture.

A network is k-partitionable if it can be divided into k isomorphic subgraphs consistent with its class definition. A partitionable network adapts better to different problem sizes, may be used to simultaneously solve several problems, and computes recursive algorithms more naturally. Since $T(h)$ is 2^i-partitionable for $i = 0...h-1$, and $MCT_r(N)$ contains N disjoint copies of $MCT_{r-1}(N)$ as subgraphs, it is easily observed that $MCT_r(N)$ is 2^{ir}-partitionable, for $i = 0...h-1$, and N^j-partitionable for $j = 0...r$.

Embedding Properties

Embedding properties are among the most important properties of a network, because they transfer the computational power of a guest network to a host network. In this section we present results regarding the embedding properties of the MCT.

Embedding the grid and the torus

We start this section by recalling a result in [2] which states that the N^r-node r-dimensional torus (and hence the grid) can be embedded into $MCT_r(N)$ with dilation 3 and congestion 2.

Although the embedding presented there does not have unit dilation and congestion, they are small and constant. Later, we will show that simple extensions made in the basic topology of the binary tree allows reducing these constants to unity. Even without such extensions, the above embeddings are much better than any embedding of the n-node grid (and, hence, the torus) in the de Bruijn or the shuffle-exchange graphs that requires $\Omega(\log \log n)$ dilation [1], or in the butterfly, cube connected cycles, or Běnes networks that require $\Omega(\log n)$ dilation [1, 6]. The slowdown introduced by these bounds in the embedding of a graph as important as the grid, reduces the practical value of these networks.

We note that the question of whether the torus or the grid can be embedded in the MOT with constant dilation and constant congestion have yet to be answered. It seems plausible, therefore, that such embeddings will probably be complex, should they exist. The simple embeddings that we have obtained seem

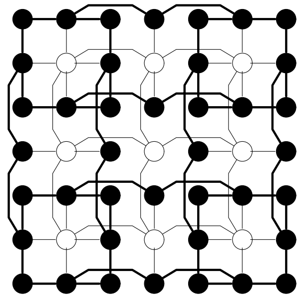

Figure 2: Embedding MOT in $MCT_2(7)$.

to indicate that the MCT does, in fact, have some advantages over the MOT network.

Combined with the above result, the next theorem establishes that the MCT is computationally more powerful than the grid.

Theorem 1 *The optimal dilation of embedding $MCT_r(N)$ in the N^r-node r-dimensional grid is $\lceil \frac{N-1}{2h-2} \rceil$.*

Proof: Section 3 in [5] presents an embedding of the N-node complete binary tree, $T(h)$, in the N-node linear array with dilation cost $\lceil \frac{N-1}{2h-2} \rceil$. The dilation of this embedding is optimal as it matches the trivial lower bound obtained by comparing the respective diameters of both networks. We can use this embedding to construct the desired embedding of the MCT in the grid by just embedding a complete binary tree onto each line of the grid.

Optimality of the dilation follows from the trivial lower bound obtained by comparing the diameters of the networks. ■

Embedding the MOT

It was shown in [2] that $MCT_r(N)$ has the r-dimensional mesh of $\frac{N+1}{2^i}$-leaf trees as subgraph, for $i = 1...h-1$ (see Figure 2). This allows us to view the MCT as a hierarchy of nested interconnected meshes of trees of different sizes, so that it may be adapted to solve multiple problems, or tailored to a particular size. When this is seen in conjunction with the facility that the embedding of the torus suggests (given the large number of algorithms developed for the guest networks considered above), we see a wide range of possible applications for the MCT.

We can show that any embedding of the mesh of N-leaf trees in the grid requires dilation $\Omega(\frac{N}{\log N})$ by comparing the respective diameters of both networks. As this dilation is large even for reasonably small values of N, The above result presents the MCT as a much better candidate than the grid for the purposes of simulating the MOT.

Embedding the complete binary tree

In [2], it is shown that the $(r(h-1)+1)$-level complete binary tree is a subgraph of $MCT_r(N)$, while any embedding of $MCT_r(N)$ in the complete binary tree requires logarithmic dilation. The combination of these two results shows that the MCT network is more powerful than the complete binary tree.

The complete binary tree subgraph of the MCT obtained in [2] is the largest possible when $r = 2$ but, in general, for bigger values of r, larger trees can be embedded with constant dilation and constant congestion. The following theorem presents this fact.

Theorem 2 *$T(rh - \lfloor \frac{r}{2} \rfloor)$ can be embedded in $MCT_r(N)$, where $N > 3$, with dilation 3 and congestion 3.*

Proof:(Sketch) It can be shown that $T(3r - \lfloor \frac{r}{2} \rfloor)$ can be embedded in $MCT_r(7)$ with dilation 3 and congestion 3 such that the root of the embedded tree is the root of $MCT_r(7)$. Note that if we remove the 2 lowest levels from every tree along each dimension in $MCT_r(N)$ we obtain a graph isomorphic to $MCT_r(2^{h-2}-1)$. Similarly, if we remove the $h-3$ top levels from every tree along each dimension we obtain a graph formed by $2^{r(h-3)}$ disjoint copies of $MCT_r(7)$. Both graphs have exactly $2^{r(h-3)}$ common nodes, that are the leaves of the $MCT_r(2^{h-2}-1)$ graph and the roots of the copies of $MCT_r(7)$ in the other graph.

In [2] was shown that $MCT_r(N)$ has a subgraph isomorphic to $T(r(h-1)+1)$. By construction, the leaves of this tree are also leaves of $MCT_r(N)$. Then, $MCT_r(2^{h-2}-1)$ has a subgraph isomorphic to $T(r(h-3)+1)$ whose leaves are the leaves of $MCT_r(2^{h-2}-1)$. At the same time, we can embed $T(3r - \lfloor \frac{r}{2} \rfloor)$ with congestion 3 and dilation 3 in each copy of $MCT_r(7)$ so that the root of the tree is the root of the copy. Combining these two embeddings we obtain the desired embedding. ■

The complete binary tree that the above theorem embeds in $MCT_r(N)$ is the largest possible for $r \leq 3$ and very close to the largest (when not the largest) for small values of r. For instance, $MCT_9(7)$ has enough nodes to contain a 25-level complete binary tree and the above theorem embeds a 23-level tree.

The case $N = 3$ is not considered in Theorem 2 although it is specially interesting because $MCT_r(3)$ is isomorphic to the r-dimensional 3^r-node grid. The result in [2] allows to obtain a complete binary tree subgraph of $MCT_r(3)$ that is the largest possible for $r \leq 3$. For larger values of r it is possible to apply an approach similar to the one used above.

Routing Properties

Shortest Path Routing:

In [10] algorithms to obtain a shortest path in a product network from the shortest path algorithms of the factor networks are given. This section studies the specific algorithm for the MCT.

The algorithm to find a shortest path in a complete binary tree can be found in [7]. The algorithm to find the shortest path in $MCT_r(N)$ is a simple extension of this algorithm for $T(h)$. The shortest path from any node x to any node y in $MCT_r(N)$ is obtained by simply applying the shortest path routing algorithm described for the tree $T(h)$ along each dimension where the labels of x and y differ.

Several shortest paths may be found if we apply the above algorithm to the dimensions in different orders. The algorithm can be implemented either in a centralized or in a distributed way. In either case, the number of time steps taken by the algorithm is proportional to the number of edges traversed by the path, since the decision process in each vertex along the path takes a constant time. Therefore any execution of the algorithm takes at most time $O(2r(h-1))$.

Broadcasting:

The broadcasting of a message is the process of sending a message from a given node to every node in the network. In [10], an algorithm is presented to broadcast a message in a product network under the multiport model. The algorithm uses the broadcast algorithms of the factor networks. The process takes at most $2r(h-1)$ steps and the algorithm can be implemented centralized or distributed. If the root of the MCT is the source of the broadcast operation then the algorithm can be completed in half the time.

Parallel Paths:

The next result establishes the number of disjoint paths that exist between any two nodes in the network. This result improves the result obtained by the direct application to our network of Theorem 2 in [3].

Theorem 3 *Every pair of vertices in $MCT_r(N)$, where $r > 1$, is connected by exactly m vertex-disjoint paths, where m is the minimum vertex degree of the vertices in the pair.*

Proof:(Sketch) We proceed by induction on the number of dimensions. For two dimensions the claim can be verified from Figure 1.a. Suppose the claim is true for $r-1$ dimensions. The case for r dimensions follows by observing that each additional dimension increases the vertex degrees by at least one and at most 3. Moreover, there is a new parallel path for each increment of the vertex degree. ∎

Corollary 1 *Every pair of vertices in $MCT_r(N)$ is connected by at least r and at most $3r$ edge-disjoint and vertex-disjoint paths.*

Corollary 2 *If $MCT_r(N)$ contains less than r faulty vertices and less than r faulty edges, it is possible to find a path between any two fault-free vertices.*

In a vertex-faulty environment the adaptive routing algorithm presented in [3] may be used with simple changes in the MCT.

Layout Properties

We now focus on the layout area bounds for the MCT network.

Theorem 4 *$MCT_r(N)$ can be laid out in an area of $\Theta(N^{2(r-1)})$ for $r > 2$ and $\Theta(N^2 \log^2 N)$ for $r = 2$.*

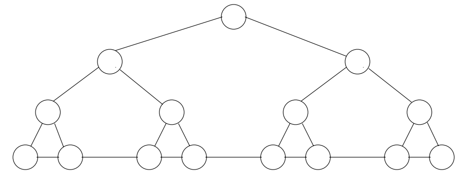

Figure 3: Extending the complete binary tree by connecting the leaves.

Proof:(Sketch) The lower bounds are implied by the fact that the largest MOT subgraph of $MCT_r(N)$ has $\Theta(N^r)$ nodes. To prove the upper bounds we show that $MCT_r(N)$ is strongly $O(x^{\frac{r-1}{r}})$-separable and then apply Theorem 3.5 in [9]. A n-node graph is said to be strongly $f(x)$-separable either if it has only one node or if by removing at most $f(n)$ edges it is divided into two graphs with the same number of nodes (within one), both strongly $f(x)$-separable. $MCT_r(N)$ can be separated into 2^r subgraphs, each composed by a $MCT_r(\frac{N-1}{2})$ graph and several isolated nodes in r steps that respect the strong separability definition. Each step removes the edges incident to the roots of a given dimension tree and distribute those roots evenly between the obtained isolated subgraphs. ∎

If we denote the number of nodes of $MCT_r(N)$ as n, the above bounds can be rewritten as $\Theta(n^{2(r-1)/r})$, for $r > 2$, and $\Theta(n \log^2 n)$, for $r = 2$. These bounds are the same as those obtained for the MOT and for the grid with more than two dimensions [9], which reflects that there is no additional asymptotical complexity cost incurred for the increased power of the MCT.

Extensions of the Basic Network

Consider connecting the leaves of the complete binary tree as shown in Figure 3. We denote the resulting graph as $XT(h)$ which is a subgraph of the X-tree graph [6]. In a modular implementation, all the nodes could be designed with the same number of I/O channels, and the unused channels at the leaves could be used to connect the leaves in this fashion. Moreover, the extra channels at the roots can be used for I/O with the external world. The next result shows that if we construct the product of these trees, denoted $MCXT_r(N)$, the resulting network has all the properties of the MCT and also contains the torus (and hence the grid) as a *subgraph*. This is a much better result than the one in Theorem 3.

Theorem 5 *$MCXT_r(N)$ contains the N^r-node r-dimensional torus as a subgraph.*

Proof: We show first that $XT(h)$ contains a Hamiltonian cycle. The claim then follows from a result in [2] which states that r-dimensional product of G_1 is a subgraph of r-dimensional product of G_2 if and only if G_1 is a subgraph of G_2.

We first show that $XT(h)$ contains the following Hamiltonian paths

LL-path: A path from the leftmost leaf to the rightmost leaf.

LR-path: A path from the leftmost leaf to the root.

Note that $XT(h)$ is symmetric and a LR-path can be converted into a path from the rightmost leaf to the root (symmetric LR-path).

We proceed by induction on h. For $h = 2$, $XT(2)$ is just a triangle and the above paths are contained in it. Therefore assume that these paths exist in $XT(h-1)$, where $h > 1$.

The LL-path for $XT(h)$ is obtained as: the LR-path in the left subtree of the root, followed by the root, followed by the symmetric LR-path in the right subtree.

The LR-path for $XT(h)$ is obtained as: the LL-path in the left subtree of the root, followed by the LR-path in the right subtree, followed by the root.

The Hamiltonian cycle for $XT(h)$ is, then, obtained as: LR-path in the right subtree of the root, followed by the root, followed by the symmetric LR-path in the left subtree. ■

Conclusions

In this paper we have analyzed many important properties of a new network called the mesh connected trees (MCT). We have shown that the mesh connected trees network has several interesting structural characteristics, including simplicity of growth, logarithmic diameter, a practically fixed vertex degree, and a large bisection width. The fixed vertex degree in particular is advantageous when compared to the hypercube, whose vertex degree grows logarithmically with increasing size. Although the MCT network is not more powerful than the hypercube, its easier implementation and significant computational power make it more desirable for certain applications.

We have shown that the MCT network is capable of hosting the torus, the grid, the mesh of trees, and the complete binary tree with small constant dilation and small constant congestion. This has considerable value because of the large number of algorithms that have been developed for these networks [6]. This capability also favors the mesh connected trees in comparison to many of the other well known networks like the shuffle-exchange, de Bruijn, butterfly, cube connected cycles, and the Běnes networks which cannot host the grid with constant dilation [1, 6].

Additional desirable properties of the MCT include efficient routing algorithms, large bandwidth, and low diameter. Finally these advantages are obtained without any increase in the asymptotical complexity of the layout area bounds over the grid (with at least 3 dimensions) or the MOT (with any number of dimensions), and with significantly less area complexity than the shuffle-exchange and de Bruijn graphs.

When we extend the basic tree topology by connecting the leaves with a straight line, we were able to show that the resulting tree contains a Hamiltonian cycle. This implies that product networks of extended trees contain tori as a subgraph, further improving the computational power of the basic architecture.

There are several features of the network that have not yet been explored. Embedding of other networks in the MCT should be considered. The relationship between the MCT and other families of networks, such as butterflies or shuffle-exchange networks has to be studied. Finally, development of algorithms for the MCT that make use of its specific topology may be investigated.

References

[1] S. N. Bhatt, F. R. K. Chung, J.-W. Hong, F. T. Leighton, B. Obrenić, A. L. Rosenberg, and E. J. Schwabe, "Optimal Emulations by Butterfly-Like Networks," *J. ACM*, 1994. to appear.

[2] K. Efe and A. Fernández, "Products of Networks with Logarithmic Diameter and Fixed Degree," *IEEE Transactions on Parallel and Distributed Systems*, 1994. Accepted for publication. Also available as Technical Report 93-8-1, CACS, University of SW. Louisiana, Lafayette, LA, Feb. 1993.

[3] T. El-Ghazawi and A. Youssef, "A Unified Approach to Fault-Tolerant Routing," in *Proceedings of the 12th International Conference on Distributed Computing Systems*, (Yokohama, Japan), pp. 210–217, June 1992.

[4] E. Ganesan and D. K. Pradhan, "The Hyper-deBruijn Networks: Scalable Versatile Architecture," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, pp. 962–978, Sept. 1993.

[5] R. Heckmann, R. Klasing, B. Monien, and W. Unger, "Optimal Embeddings of Complete Binary trees into Lines and Grids," in *Proceedings of 17th International Workshop, WG'91, Graph-Theoretic Concepts in Computer Science* (G. Schmidt and R. Berghammer, eds.), vol. 570 of *Lecture Notes in Computer Science*, pp. 25–35, Fischbachau, Germany: Springer Verlag, June 1991.

[6] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, and Hypercubes*. San Mateo: Morgan Kaufmann, 1992.

[7] D. K. Pradhan, ed., *Fault-Tolerant Computing: Theory and Techniques*, vol. 2, ch. 7. Englewood Cliffs: Prentice Hall, 1986.

[8] A. L. Rosenberg, "Product-Shuffle Networks: Toward Reconciling Shuffles and Butterflies," *Discrete Applied Mathematics*, vol. 37/38, pp. 465–488, July 1992.

[9] J. D. Ullman, *Computational Aspects of VLSI*. Rockville: Computer Science Press, 1984.

[10] A. Youssef, "Product Networks: A Unified Theory of Fixed Interconnection Networks," Tech. Rep. GWU-IIST-90-38, Institute for Information Science and Technology, The George Washington University, Washington, D.C., Dec. 1990.