Telecommunication Systems 13 (2000) 21–27 

# The Shuffled Mesh: a flexible and efficient model for parallel computing

G. Bongiovanni, G.A. De Biase, A. Massini and A. Monti
*Dipartimento di Scienze dell'Informazione, Università di Roma la Sapienza, Via Salaria 113, 00198 Roma, Italy*

In this work an efficient model for parallel computing, called Shuffled Mesh (SM), is introduced. This bounded degree model has the mesh as subgraph and it is based on the union of mesh and shuffle-exchange topologies. It is shown that an $N$-processor SM combines the features of mesh, shuffle-exchange, hypercubic networks, mesh of trees and hypercube, and is able to support all the algorithms designed for such topologies with constant or logarithmic time performance degradation. Finally, it is proved that the VLSI layout of a SM is the same as of a shuffle exchange of the same size.

## 1. Introduction and preliminaries

Over the last years all the aspects of parallel computing have received considerable attention in the literature. The main reasons for this growing interest are the difficulties to increase the performance of sequential computers due to some physical limitations. A collection of processors working in parallel can increase computing performance, and it can therefore become suitable to solve problems in many areas of science and engineering. Because several technical difficulties prevent the realization of PRAM-like multiprocessor architectures with very large $N$ (number of processors), parallel computing models with various topologies have been studied. The most popular models are arrays (meshes), mesh of trees, hypercube, shuffle-exchange and other hypercubic networks. Unfortunately, every topology (multiprocessor architecture) presents some limitations, which do not allow to solve different classes of problems with the same efficiency.

Arrays are the simplest topologies for parallel computation. Two-dimensional arrays (meshes) are useful for image processing, computational geometry, graph algorithms, etc. Meshes are scalable and relatively simple to build, but they suffer from a major drawback: their large *diameter* (the maximum distance between any pair of processors) which limits their computing speed on many problems.

To overcome this fact a hybrid topology based on mesh and trees (the mesh of trees) was formally defined [7]. This fast topology has a smaller diameter, but requires $2N^2 - 2N$ additional nodes for the information routing. The typical running times of $\Theta(\sqrt{N})$ or $\Theta(N)$ of algorithms on meshes become $\Theta(\log N)$ and $\Theta(\log^2 N)$ (often without increasing the number of processors) [9].

The hypercube is one of the most versatile and efficient topology used for parallel computation. It can efficiently simulate any other topology of the same size $N$ and of the same wire complexity. It has the mesh as subgraph [2] and can simulate the mesh of trees with only a small constant slowdown [4]. The hypercube is quite powerful from a computational point of view, but it presents some disadvantages (one disadvantage is that it is not scalable, since its node degree grows with its size).

To overcome this problem several bounded-degree hypercube derivatives (*hypercubic networks*) have been proposed and studied. The most popular derivatives are the shuffle-exchange [11], the butterfly, the omega network, etc., and the cube-connected-cycles [10]. All these networks are virtually identical from a computational standpoint [9] and are universal (i.e., they can simulate any other network with a comparable number of processors and wires with only a logarithmic factor slowdown [9]). An $N$-node hypercubic network can simulate any computation of an $N$-node hypercube with only a constant factor loss in efficiency for any *normal* hypercube algorithm (algorithms in which only one dimension of the hypercube edges is used at any step, and consecutive dimensions are used in consecutive steps – as is almost always the case [9]), while all hypercube algorithms can be implemented on these networks with $O(\log^2 N)$ slowdown.

In this work, in order to achieve a topology able to efficiently simulate meshes, shuffle exchange, hypercubic networks, hypercube and others parallel computing models, the *Shuffled Mesh* (SM) topology is proposed. This model has been used in the past by Chung and Lin to obtain a cost-optimal parallel algorithm for B-spline surface fitting [3]. The SM is obtained by combining the structure of the mesh with the structure of the shuffle-exchange. It is a bounded degree topology which trivially embeds the mesh and can efficiently simulate (with only a small constant slowdown) a shuffle-exchange of the same size. The mesh of trees can be efficiently simulated on the SM model, recalling that a $2N - 1$ complete binary tree may be embedded with constant dilation, congestion and load in an $N$-shuffle-exchange [9].

Hence, all the algorithms designed for the mesh run on the SM without slowdown, algorithms designed for the hypercubic networks run with a constant slowdown, and consequently normal hypercube algorithms can be implemented on the SM with the same slowdown. Finally, it is proved that the SM requires $\Theta(N^2/\log^2 N)$ area, the same as the shuffle exchange and all hypercubic networks [5,12].

## 2. The Shuffled Mesh model

The graph describing the $N$-node mesh topology has $N = s \times s$ nodes. It is the graph whose vertices comprise the ordered pairs of integers $\{0, \ldots, s-1\} \times \{0, \ldots, s-1\}$ and whose edges connect vertices $(a, b)$ and $(c, d)$ just when $|a-c|+|b-d| = 1$.

The graph describing the $N$-node shuffle-exchange topology (see figure 1) has $N = 2^k$, $k \geqslant 1$, nodes. Each node corresponds to a unique $k$-bit binary number, and two nodes $u$ and $v$ are linked by an edge if either: (a) $u$ and $v$ differ in precisely the

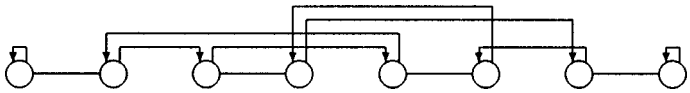

Figure 1. The 8-node shuffle-exchange.

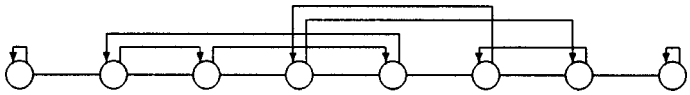

Figure 2. The 8-node shuffle-shift.

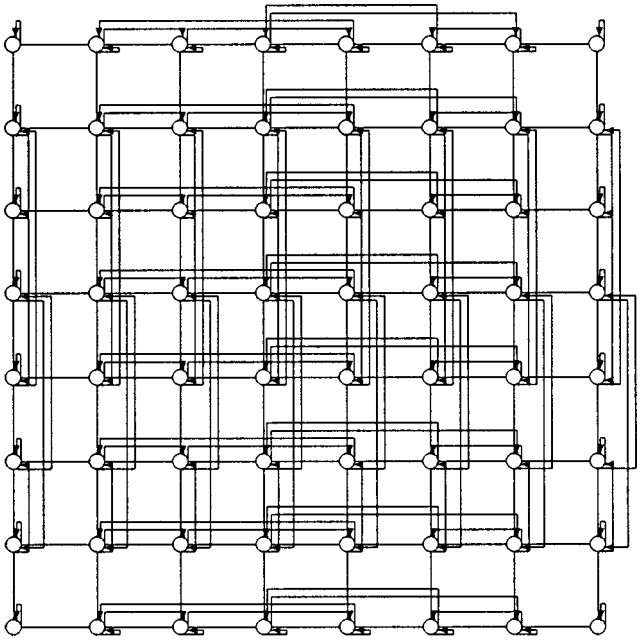

Figure 3. The 64-node Shuffled Mesh.

last bit, or (b) $u$ is a left or right cyclic shift of $v$. If $u$ and $v$ differ in the last bit, the edge is called an *exchange* edge. Otherwise, the edge is called a *shuffle* edge.

An $N$-node *shuffle-shift* graph [5] (see figure 2) is an $N$-node shuffle-exchange graph where shift edges (i.e., edges linking the $i$th node to the $(i+1)$st node, for all odd $i$) have been added.

Now the $N$-node Shuffled Mesh topology is a 2-dimensional shuffle-shift. Its graph, as shown in figure 3, is the union of an $N$-node mesh and $\sqrt{N}$ (vertical) $\times\sqrt{N}$ (horizontal) shuffle-exchanges, each with $\sqrt{N}$ nodes. The graph describing an $N$-node SM has $N = 2^{2k}$ nodes and it is obtained from an $N$-node mesh by adding shuffle edges to each row and each column. Each node of the SM corresponds to a unique $2k$-bit binary number, the node in the $j$th column of the $i$th row corresponds to bin($i$)|bin($j$) ($0 \leqslant i, j < \sqrt{N}$), where bin($i$)|bin($j$) denotes the binary $k$-bit representa-

tion of $i$ concatenated with the $k$-bit binary representation of $j$. As shown in figure 3, the SM is a bounded 8-degree structure.

Following [8], let an *embedding* of a graph $G$ into a graph $H$ be a mapping $\phi : G \to H$ that maps the nodes of $G$ onto the nodes of $H$, and the edges of $G$ onto the paths in $H$, let the *dilation* of an embedding, $d$, be the length of the longest path $\phi(e)$ corresponding to an edge of $G$, and let the *congestion* of an embedding, $c$, be the largest number of paths $\phi(e)$ corresponding to a single edge of $H$. The *load* of an embedding, $l$, is the maximum number of nodes of $G$ mapped to a single node of $H$ (e.g., in a one-to-one embedding the load is 1), and if there is a $c$-congested, $d$-dilated, and $l$-loaded embedding of $G$ in $H$, then there is an emulation of $G$ by $H$ with slowdown O($l + c + d$).

**Theorem 1.** Any $N$-node shuffle-exchange can be one-to-one embedded in an $N$-node Shuffled Mesh with 4 dilation and 2 congestion.

*Proof.* Let $N = 2^{2k}$. In the following, for any $2k$-bit binary number $w$, $(w)$ denotes the corresponding node on the $N$-shuffle-exchange, and $\langle w \rangle$ denotes the corresponding node on the $N$-node SM. By assigning $(a_0 \ldots a_{k-1} b_0 \ldots b_{k-1})$ to $\langle a_0 \ldots a_{k-1} b_0 \ldots b_{k-1} \rangle$, the shuffle-exchange can be one-to-one embedded in the SM. Any exchange edge of the shuffle-exchange $(a_0 \ldots a_{k-1} b_0 \ldots b_{k-1}) \to (a_0 \ldots a_{k-1} b_0 \ldots \overline{b_{k-1}})$ to the path of length one (i.e., edge) in the SM connecting $\langle a_0 \ldots a_{k-1} b_0 \ldots b_{k-1} \rangle$ to $\langle a_0 \ldots a_{k-1} b_0 \ldots \overline{b_{k-1}} \rangle$, can be trivially mapped. Now consider the shuffle edge connecting $(a_0 a_1 \ldots a_{k-1} b_0 b_1 \ldots b_{k-1})$ to $(a_1 \ldots a_{k-1} b_0 \ldots b_{k-1} a_0)$:

- When $a_0 = b_0$, the edge is mapped to the following path of length 2:

$$\begin{aligned}
&\langle a_0 a_1 \ldots a_{k-1} b_0 b_1 \ldots b_{k-1} \rangle \to \\
&\langle a_1 \ldots a_{k-1} a_0 b_0 b_1 \ldots b_{k-1} \rangle \to \\
&\langle a_1 \ldots a_{k-1} a_0 b_1 \ldots b_{k-1} b_0 \rangle = \langle a_1 \ldots a_{k-1} b_0 b_1 \ldots b_{k-1} a_0 \rangle.
\end{aligned}$$

- When $a_0 \neq b_0$, the edge is mapped to the following path of length 4:

$$\begin{aligned}
&\langle a_0 a_1 \ldots a_{k-1} b_0 b_1 \ldots b_{k-1} \rangle \to \\
&\langle a_1 \ldots a_{k-1} a_0 b_0 b_1 \ldots b_{k-1} \rangle \to \\
&\langle a_1 \ldots a_{k-1} a_0 b_1 \ldots b_{k-1} b_0 \rangle \to \\
&\langle a_1 \ldots a_{k-1} a_0 b_1 \ldots b_{k-1} \overline{b_0} \rangle \to \\
&\langle a_1 \ldots a_{k-1} \overline{a_0} b_1 \ldots b_{k-1} \overline{b_0} \rangle = \langle a_1 \ldots a_{k-1} b_0 b_1 \ldots b_{k-1} a_0 \rangle.
\end{aligned}$$

□

From theorem 1 follows that the $N$-node shuffle-exchange graph can be one-to-one embedded in the $N$-node SM graph with 4-dilation and constant congestion. All the exchange connections have been mapped in the SM edge-disjoint paths, and in every path of length 4 of the shuffle edges, the 3rd edge is shared with the edge for an exchange, that implies an embedding with 2-congestion.

**Corollary 1.** The $N$-node Shuffled Mesh can simulate an $N$-node shuffle-exchange with constant slowdown.

Thus, the SM is universal, it can simulate all the hypercubic networks with only constant slowdown. Any normal hypercube algorithm can be implemented on this model with a constant factor loss in efficiency and any hypercube algorithm can be implemented on this model with $O(\log^2 N)$ loss in efficiency.

## 3. The Shuffled Mesh VLSI layout

**Lemma 1.** The $N$-node Shuffled Mesh requires $\Omega(N^2/\log^2 N)$ area as the shuffle-exchange.

*Proof.* Since the $N$-node shuffle-exchange graph can be one-to-one embedded in the $N$-node SM with constant congestion, we can simply blowup any SM layout by a constant factor to obtain layouts for the shuffle-exchange graph with equivalent area. The $N$-node shuffle-exchange graph requires $\Omega(N^2/\log^2 N)$ area [12], hence, the same is true for the $N$-node SM. □

To prove the subsequent lemma 2, some recalls from [5] are needed. A *necklace* of the $N$-node shuffle-shift graph is the collection of all the cyclic shifts of a node of the graph. When the size of this collection is $\log N$, the necklace is said to be *full*.

An optimal layout area, $A$, of the $N$-node shuffle-shift is obtained by two contributions $A = A_1 + A_2$. $A_1$ is given by the shuffle-shift subgraph containing only nodes belonging to full necklaces, and it requires $O(N/\log N)$ vertical and $O(N/\log N)$ horizontal tracks. The second contribution $A_2$ is given by only the $O(N^{1/2} \log N)$ nodes belonging to not full necklaces. Because $m$ wires can always be inserted into any layout with the addition of at most $2m$ vertical and $2m$ horizontal tracks, the set of $O(N^{1/2} \log N)$ disregarded nodes and edges can be inserted into $A$ without increasing the total $O(N^2/\log^2 N)$ area by more than a constant factor.

Still following [5], there exists in $A_1$ an ordering of the nodes in a grid of $\log N$ rows and $O(N/\log N)$ columns such that the insertion of all the links among such nodes with the addition of just $O(N/\log N)$ vertical and $O(N/\log N)$ horizontal tracks is possible. This result remains true when a grid of $\log N$ rows and $\log N$ columns, obtained by expanding each row of the previous grid with a $\log N \times \log N$ subgrid, is used, and the nodes of the original row are assigned on the diagonal of such subgrid. The order from left to right of the nodes in this grid will be called *canonic order*.

**Lemma 2.** The $N$-node Shuffled Mesh requires $O(N^2/\log^2 N)$ area as the shuffle-exchange.

*Proof.* All rows and columns of an $N$-node SM are $\sqrt{N}$-node shuffle-shift graphs. Each SM node belongs to two necklaces: a vertical necklace and a horizontal necklace

(see figure 3). The SM layout area, $L$, can be obtained by adding two contributions $L_1$ and $L_2$ (similarly to the shuffle-shift area). Let $G$ be the subgraph of the $N$-node SM induced by the set of all the nodes belonging to horizontal and vertical full necklaces. The nodes of $G$ belonging to the same row (or column) of the original SM are a subset of the nodes of a $\sqrt{N}$-node shuffle-shift. Dispose the nodes of $G$ in a $O(\sqrt{N}) \times O(\sqrt{N})$ grid so that all the nodes belonging to the same row (or column) of the SM result in canonic order. All the edges which link the nodes in $G$ corresponding to a row (column) of the SM can be inserted with the addition of just $O(\sqrt{N}/\log N)$ vertical and $O(\sqrt{N}/\log N)$ horizontal tracks. Since the SM has $O(\sqrt{N})$ rows and $O(\sqrt{N})$ columns, all the edges connecting nodes of G can be inserted with the addition of just $O(N/\log N)$ vertical tracks and $O(N/\log N)$ horizontal tracks. Thus, the area of $L_1$ is $O(N^2/\log^2 N)$. To estimate the contribution of $L_2$ the set of disregarded nodes must be inserted. In a $\sqrt{N}$-node shuffle-shift $O(N^{1/4} \log N)$ nodes are contained in not full necklaces, and consequentely in the $N$-node SM $O(N^{3/4} \log N)$ nodes are contained in vertical or horizontal not full necklaces. They can be added to the layout with the addition of $O(N^{3/4} \log N)$ vertical and $O(N^{3/4} \log N)$ horizontal traks. Hence, the area of $L_2$ is $O(N^{3/2} \log^2 N)$. Thus, the SM layout area $L$ is $O(N^2/\log^2 N)$ because $L_2$ increases by no more than a constant factor. □

From lemmas 1 and 2 immediately follows:

**Theorem 2.** The layout for the $N$-node Shuffled Mesh requires $\Theta(N^2/\log^2 N)$ area as the shuffle-exchange.

## 4. Concluding remarks

In the previous sections the SM model has been examined as a bounded degree 2-dimensional one-stage recirculant network. By this approach various kinds of topologies can be embedded or simulated by constant slowdown (mesh, shuffle-exchange, hypercube – only for normal algorithms – and all hypercubic networks) or logarithmic slowdown (tree based topologies like mesh of trees, pyramid) without any increase in the number of processors. Any embedding of an $N$-node mesh into the butterfly or the cube-connected-cycles requires dilation $\Omega(\log N)$ [1], and dilation $\Omega(\log \log N)$ into the shuffle-exchange [9]. Since the mesh is a subgraph of the SM, the above results hold also for the SM which cannot be embedded in the hypercubic networks with constant dilation. It is important to note that the fact that a graph cannot be embedded in the hypercubic networks with constant dilation does not necessarily mean that these networks cannot efficiently simulate this graph (e.g., the hypercubic networks can still simulate meshes with constant slowdown [6], although the details of the simulations are quite complicated and nonintuitive); this result appears to be only of theoretical interest. Hypercubic networks might still simulate the SM in an efficient way (i.e., from the computational point of view, SM and hypercubic networks might still be equivalent), but this result would still have only a theoretical interest.

The SM can also be built in non-recirculant mode reproducing the mesh $\log \sqrt{N}$ times, and connecting the nodes of subsequent stages by a shuffle-exchange structure. In this case, at present under study, an enhancement of the SM model efficiency is obtained. In fact, with this second approach, the SM has as subgraph (in addition to the mesh) the mesh of trees and the pyramid, reducing the corresponding slowdowns and reducing also the slowdown with respect to the hypercubic networks and the hypercube.

## References

[1] S.N. Bhatt, F.R.K. Chung, J.-W. Hong, F.T. Leighton and A.L. Rosemberg, Optimal simulations by butterfly networks, in: *Proc. of the 20th Annual ACM Symposium on the Theory of Computing* (1988) pp. 192–204.

[2] M. Chan, Dilation 2-embeddings of grids into hypercubes, in: *Proc. of the 1988 Internat. Conf. on Parallel Processings* (1988) pp. 295–298.

[3] K.-L. Chung and F.-C. Lin, A cost-optimal parallel algorithm for B-spline surface fitting, Computer Vission, Graphics, and Image Processing 53 (1991) 601–605.

[4] K. Efe, Embedding mesh of trees in the hypercube, Journal of Parallel and Distributed Computing 11 (1991) 222–230.

[5] D. Kleitman, F.T. Leighton, M. Lepley and G.L. Miller, New layouts for shuffle-exchange graph, in: *Proc. of the 13th Annual ACM Symposium on the Theory of Computing* (1981) pp. 278–292.

[6] R. Koch, T. Leighton, B. Maggs, S. Rao and A. Rosenberg, Work-preserving emulations of fixed-connection networks, in: *Proc. of the 21th Annual ACM Symposium on the Theory of Computing* (1989) pp. 227–240.

[7] T. Leighton, New lower bound techniques for VLSI, in: *Proc. of the 22th Annual IEEE Symposium on Foundations of Computer Science* (1981) pp. 1–12.

[8] T. Leighton, B. Maggs and S. Rao, Universal packet routing algorithms, in: *Proc. of the 29th Annual IEEE Symposium on Foundations of Computer Science* (1988) pp. 256–271.

[9] T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes* (Morgan Kaufmann, San Mateo, CA, 1992).

[10] F.P. Preparata and J.E. Vuillemin, The cube-connected cycles: A versatile network for parallel computation, in: *Proc. of the 12th Annual IEEE Symposium on Foundations of Computer Science* (1979) pp. 140–147.

[11] H.S. Stone, Parallel processing with the perfect shuffle, IEEE Transactions on Computers 20 (1971) 153–161.

[12] C.D. Thompson, A complexity theory for VLSI, Ph.D. thesis, Carnegie-Mellon University, Pittsburg, PA (1980).