HOW PROMISING IS THE k-CONSTRAINED RECONFIGURABLE MESH?

M. Manzur Murshed*
Gippsland School of Comp. & Info. Tech.
Monash University
Churchill VIC 3842, Australia
Manzur.Murshed@infotech.monash.edu.au

Richard P. Brent
Oxford University Comp. Lab
Oxford University
Oxford, OX1 3QD, U.K.
rpb@comlab.ox.ac.uk

November 4, 1999

Abstract

The Reconfigurable Mesh (RM) attracted criticism for its key assumption that a message can be broadcast in constant time independent of bus length. To account for this limit, k-constrained RM model has recently been proposed where buses of length at most k segments are allowed to be configured at any step. Straightforward simulations of AT^2 optimal RM algorithms on this constrained model are found to be non-optimal and several AT^2 optimal algorithms have already been developed on this new model. In this paper, we prove that if attaining AT^2 optimality is a goal of developing algorithms on the k-constrained RM, then the k-constrained RM does not make any difference to the original "unconstrained" RM.

1 Introduction

It is well known that inter-processor communications and simultaneous memory accesses often act as bottlenecks in present-day parallel machines. Bus systems have been introduced recently to a number of parallel machines to address this problem. Examples include the *Bus Automaton* [13], the *Reconfigurable Mesh (RM)* [10], the *content addressable array processor* [16], and the *Polymorphic torus* [9]. Among them RM draws much attention because of its simplicity. A bus system is called *reconfigurable* if it can be dynamically changed according to either global or local information.

In the most common *unit-time delay* model of RM, it is assumed that a message can be broadcast in constant time along any bus independent of its length. This

*Corresponding author.

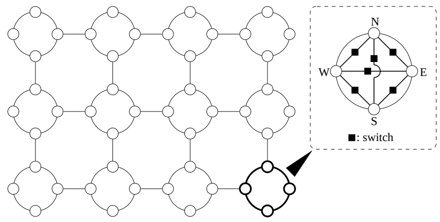

Figure 1: A reconfigurable mesh of size 3×4.

assumption attracted criticism and cast a shadow of doubt on the implementation of RM. Although investigations of bus delays in [6, 7, 8] has confirmed that broadcast delay is very small, theoretically it cannot be correctly modeled by a constant independent of bus length. To account for this limit, Beresford-Smith *et al.* [3, 2] have proposed the k*-constrained* RM where buses of length at most k segments are allowed to be formed at any time. Straightforward simulations of AT^2 optimal RM algorithms on this constrained model are found to be non-optimal and several AT^2 optimal algorithms have already been developed on this new model.

In this paper we first show that to keep a k-constrained RM of size $p \times q$ distinct from an "unconstrained" RM of same size, k must be $o(p+q)$. We then prove that to make the k-constrained RM more powerful than the ordinary mesh, k must not be a constant, i.e., k must be a function of p and q. Finally we show that if attaining AT^2 optimality is a goal of developing algorithms on the k-constrained RM, as has been discussed by Beresford-Smith *et al.* in [3, 2], then the k-constrained RM does not make any difference to the original "unconstrained" RM in considering propagation delay in the time complexity of algorithms.

Throughout the paper, we use $\Theta()$ to mean "oder exactly'," $O()$ to mean "order at most," $\Omega()$ to mean "order at least," and $o()$ to mean "order less."

The paper is organized as follows. Section 2 presents various models of reconfigurable meshes discussed in this paper. Analysis of relative power of the k-constrained RM with the ordinary mesh and the "unconstrained" RM is done in Section 3. Section 4 concludes the paper.

2 The Reconfigurable Mesh Models

The reconfigurable mesh is primarily a two-dimensional mesh of processors connected by reconfigurable buses. In this parallel architecture, a processor element is placed at the grid points as in the ordinary mesh connected computers. Processors of the

RM of size $X \times Y$ are denoted by $PE_{i,j}$, $0 \leq i < X$, $0 \leq j < Y$ where processor $PE_{0,0}$ resides in the south-western corner. Each processor is connected to at most four neighboring processors through fixed bus segments connected to four I/O ports **E** & **W** along dimension x and **N** & **S** along dimension y. These fixed bus segments are building blocks of larger bus components which are formed through switching, decided entirely on local data, of the internal connectors (see Figure 1) between the I/O ports of each processor.

Other than the buses and switches the RM of size $p \times q$ is similar to the standard mesh of size $p \times q$ and hence it has $\Theta(pq)$ area in VLSI embedding [14], under the assumption that processors, switches, and links between adjacent switches occupy unit area.

One critical factor in the complexity analysis of reconfigurable algorithms is the time needed to propagate a message over a bus. In the most common *unit-time delay* model [15], it is assumed that in any configuration any message can be transmitted along any bus in constant time, regardless of the bus length. Unfortunately this assumption, based on which a large number of algorithms with constant time complexity are developed, is theoretically false, as the speed of signals carrying information is bounded by the speed of light. This partially explains why the reconfigurable meshes have not gained wide acceptance initially. Recently some VLSI implementations of reconfigurable meshes have demonstrated that the broadcast delay, though not a constant, is nevertheless relatively small in terms of machine cycles. For example, only 16 machine cycles are required to broadcast on a 10^6 processor YUPPIE (Yorktown Ultra Parallel Polymorphic Image Engine) [7, 8]. GCN (Gated-Connection Network) [6] has even shorter delays by adopting precharged circuits. Broadcast delay can further be reduced by using optical fiber for reconfigurable bus system and electrically controlled directional coupler switches as proposed in [1].

Although the above observations serve the practical purposes, unit-time delay can never be theoretically sound. In the *log-time delay* model [11] it is assumed that each broadcast takes $\Theta(\log s)$ time to reach all the processors connected to a bus, where s is the maximum number of switches in a minimum switch path between two processors connected on the bus. The log-time delay model accounts for the delay and therefore, it is more realizable approach in analyzing algorithms' complexity. But in terms of the speed of light it is also not a very realistic model.

Very recently a different approach is considered to account for the delay associated with propagation. Beresford-Smith *et al.* [2, 3] have recently proposed the *k-constrained* RM model where it is assumed that in any situation any message can propagate at most k fixed bus segments and thus buses of length at most k segments are allowed in any step.

Although Beresford-Smith *et al.* [2, 3] have not proposed any upper limit on the value of k, it is obvious that a k-constrained RM of size $p \times q$ should have $k = o(p+q)$ for large $p+q$. Otherwise, the k-constrained RM model will be asymptotically same as the "unconstrained" (unit-time delay model) RM and therefore, the delay associated with propagation will remain unaccounted for.

3 Relative Power of the k-Constrained RM

Definition 1 *Let M_1 and M_2 denote two parallel computational models. Model M_1 is considered* as powerful as *model M_2 if and only if each step in M_2 with N processors can be simulated in constant time by M_1 with $O(N^{\varepsilon})$ processors where ε is a small constant ≥ 1. Model M_1 is considered* more powerful than *model M_2 if and only if M_1 is as powerful as M_2 but M_2 is not as powerful as M_1.*

Theorem 1 *The k-constrained RM is as powerful as the ordinary mesh.*

Proof. The k-constrained RM can configure buses of length at most k units. As the ordinary mesh uses only unit-lengthed buses, a k-constrained RM of size $p \times q$ can execute any algorithm on an ordinary mesh of size $p \times q$ in the same time complexity without any modification in the algorithm. ■

Theorem 2 *The ordinary mesh is as powerful as the k-constrained RM, where k is a constant.*

Proof. Consider an arbitrary step of an arbitrary algorithm on a k-constrained RM of size $p \times q$. This step configures buses of length at most k units. Now an ordinary mesh of size $p \times q$ can simulate each of these buses of length at most k units in $O(k)$ times. As k is a constant, $O(k)$ can be considered as $O(1)$, i.e., constant time. ■

Theorem 3 *The k-constrained RM, where k is a constant, is not more powerful than the ordinary mesh.*

Proof. A proof of contradiction can easily be arranged using Theorem 2. ■

So, the above Theorem 3 clearly proves that in the definition of the k-constrained RM of size $p \times q$, where $k = o(p+q)$, k must not be considered as a constant and therefore, k must always be considered as a function of p and q. The same observation has also been made by Beresford-Smith *et al.* [2, 3] without any formal proof.

Theorem 4 *The RM is as powerful as the k-constrained RM.*

Proof. As the RM can configure buses of any length, a RM of size $p \times q$ can execute any algorithm on a k-constrained RM of size $p \times q$ in the same time complexity without any modification in the algorithm. ■

Theorem 5 *The k-constrained RM is not as powerful as the RM.*

Proof. Consider solving a problem $\mathcal{P}$ of size n in constant time and suppose this demands an RM of size at least $p \times q$. By Theorem 4, no k-constrained RM of size less than $p \times q$ should be able to solve the same problem $\mathcal{P}$ of size n in constant time.

Now, as stated at the end of Section 2, for large $p + q$, k must be assumed as $o(p + q)$. Otherwise, the k-constrained RM model will be asymptotically same as the "unconstrained" RM. So, no k-constrained RM can solve problem $\mathcal{P}$ of size n in constant time as the communication diameter of the minimum allowable sized k-constrained RM is $(p + q)/o(p + q) \neq O(1)$. ■

The consequence of Theorem 5 has also been realized by Beresford-Smith *et al.* [2, 3] as they have shown, in obvious way, that straightforward simulation of an RM algorithms on a k-constrained RM loses AT^2 optimality unless the area of the mesh is reduced. To address this issue, AT^2 optimal algorithms have already been developed on k-constrained RMs for sorting and computing convex hull [2, 3], broadcasting [4], multiplying sparse matrices [5], and computing the contour of maximal elements of a set of planar points [12].

By admitting the importance of AT^2 optimality and then developing a number of AT^2 algorithms on the k-constrained RM, Beresford-Smith *et al.* [2, 3], ironically, have also revealed a fatal weakness in their model. In the rest of this section we show that if attaining AT^2 optimality is a goal of developing algorithms on the k-constrained RM, difference between this model and the "unconstrained" model disappears.

Let a problem $\mathcal{P}$ of size n have $I(n)$ *information content* [14, pp. 51–54]. If this problem $\mathcal{P}$ is realized in a VLSI circuit with aspect ratio $\alpha \geq 1$ then, by Ullman [14, pp. 57], the AT^2 lower bound of $\mathcal{P}$ will be

$$\Omega(\alpha I^2(n)) \; . \tag{1}$$

Now, consider a mesh of size $p \times q$ where $pq = \mathcal{K}I(n)$, $1 < p \leq q \leq I^2(n)$, and $\mathcal{K} \geq 1$.

We assume that initially each item of $I(n)$ information content is contained in a distinct processor. In such case $\mathcal{K} = \frac{pq}{I(n)}$ has a physical interpretation. It represents how much of the mesh is filled with data.

Let $\mathcal{P}$ be solved, AT^2 optimally, on a mesh of size $p \times q$ in $O(T)$ time. Then applying eqn. (1) we get

$$pqT^2 = \frac{q}{p} I^2(n) \ .$$

This implies

$$T = \frac{I(n)}{p} = \frac{q}{\mathcal{K}} \ . \tag{2}$$

No matter what type of mesh (ordinary or reconfigurable) we are using and no matter whether propagation delay is accounted for (log-time delay and k-constrained RM models) or not ("unconstrained" unit-time delay RM model), to attain AT^2 optimality, problem $\mathcal{P}$ must be solved in $O(q/\mathcal{K})$ time on a mesh of size $p \times q$ where $pq = \mathcal{K}I(n)$ and $p \leq q$.

If the above mesh is an ordinary one, then $\mathcal{K}$ must be $O(1)$ as the communication diameter is $\Theta(q)$. If the mesh is a k-constrained RM, then $\mathcal{K}$ must be $O(k)$ as the communication diameter is $\Theta(q/k)$. $\mathcal{K}$ can take any possible value if the mesh under consideration is an "unconstrained" RM as the communication diameter is 1.

So, given a problem of fixed size to be solved in a k-constrained RM of fixed size, attaining AT^2 optimality depends on the factor $\mathcal{K}$. If $\mathcal{K} > k$ then we must reduce the size of the mesh so that $\mathcal{K} = \frac{pq}{I(n)} = k$. This means to solve a problem $\mathcal{P}$ of size n with $I(n)$ information content on a k-constrained RM of size $p \times q$, the area of the mesh pq must be $O(kI(n))$.

Theorem 6 *It is possible to solve problems AT^2 optimally on the k-constrained RM if the area of the mesh is unconstrained.* ∎

Let a problem $\mathcal{P}$ of size n with $I(n)$ information content is solved by algorithm $\mathcal{A}$, AT^2 optimally, on a k-constrained RM of size $p \times q$, $pq = kI(n)$. Suppose the same problem $\mathcal{P}$ of size n with $I(n)$ information content now needs to be solved AT^2 optimally using an "unconstrained" RM of size $p \times q$. By Theorem 4, the same algorithm $\mathcal{A}$, without making any modification, can be used for this purpose. So, even with the "unconstrained" model, it is sufficient to construct buses of length at most k segments.

Theorem 7 *If attaining AT^2 optimality is a goal of developing algorithms on the k-constrained RM, the difference between this model and the "unconstrained" unit-time delay RM model, in considering propagation delay in measuring time complexity of algorithms, is lost.* ■

4 Conclusion

The Reconfigurable Mesh (RM) attracted criticism for its key assumption that a message can be broadcast in constant time independent of bus length. To account for this limit Beresford-Smith *et al.* [3, 2] have recently proposed the k-constrained RM model where buses of length at most k segments are allowed to be configured at any step. Straightforward simulations of AT^2 optimal RM algorithms on this constrained model are found to be non-optimal and several AT^2 optimal algorithms have already been developed on this new model. In this paper we first have shown that to keep a k-constrained RM of size $p \times q$ distinct from an "unconstrained" RM of same size, k must be $o(p+q)$. Then we have proved that to make the k-constrained RM more powerful than the ordinary mesh, k must not be a constant, i.e., k must be a function of p and q. Finally we have established that if attaining AT^2 optimality is a goal of developing algorithms on the k-constrained RM then the k-constrained RM does not make any difference to the original "unconstrained" RM in considering propagation delay in the time complexity of algorithms.

References

[1] Y. Ben-Asher, D. Peleg, R. Ramaswami, and A. Schuster. The power of reconfiguration. *Journal of Parallel and Distributed Computing*, 13:139–153, 1991.

[2] B. Beresford-Smith, O. Diessel, and H. ElGindy. Optimal algorithms for constrained reconfigurable meshes. *Australian Computer Science Communications*, 17:32–41, 1995.

[3] B. Beresford-Smith, O. Diessel, and H. ElGindy. Optimal algorithms for constrained reconfigurable meshes. *Journal of Parallel and Distributed Computing*, 39:74–78, 1996.

[4] O. Diessel, H. ElGindy, and L. Wetherall. Efficient broadcasting procedures for constrained reconfigurable meshes. Technical Report 96-07, Department of Computer Science, The University of Newcastle, Australia, 1996.

[5] H. ElGindy. On sparse matrix multiplication for constrained reconfigurable meshes. In *Proceedings of the 3rd Workshop on Reconfigurable Architectures*, Honolulu, Hawaii, April 1996.

[6] H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *J. ACM*, 22:469–476, 1975.

[7] H. Li and M. Maresca. Polymorphic-torus network. *IEEE Transactions on Computers*, 38:1345–1351, 1989.

[8] R. Lin, S. Olariu, J. Schwing, and J. Zhang. A VLSI-optimal constant time sorting on reconfigurable mesh. In *Ninth European Workshop Parallel Computing*, pages 1–16, Spain, 1992.

[9] M. Maresca. Polymorphic processor arrays. *IEEE Transactions on Parallel and Distributed Systems*, 4:490–506, 1993.

[10] R. Miller, V. K. P. Kumar, D. I. Reisis, and Q. F. Stout. Data movement operations and applications on reconfigurable VLSI arrays. In *Proc. International Conference on Parallel Processing*, pages 205–208, 1988.

[11] R. Miller, V. K. Prasanna-Kumar, D. I. Reisis, and Q. F. Stout. Parallel computations on reconfigurable meshes. *IEEE Transactions on Computers*, 42:678–692, 1993.

[12] M. M. Murshed and R. P. Brent. Maximal contour algorithms on constrained reconfigurable meshes. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99)*, volume IV, pages 2238–2244, Las Vegas, Nevada, U.S.A., June 28–July 1 1999.

[13] J. Rothstein. Bus automata, brains, and mental models. *IEEE Trans. Syst. Man Cybern*, 18:522–531, 1988.

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

[15] B.-F. Wang and G.-H. Chen. Constant time algorithms for the transitive closure and some related graph problems on processor arrays with reconfigurable bus systems. *IEEE Transactions on Parallel and Distributed Systems*, 1:500–507, 1990.

[16] C. C. Weems et al. The image understanding architecture. *Internat. J. of Comput. Vision*, 2:251–282, 1989.