Quadrisection-Based Task Mapping on Many-Core Processors for Energy-Efficient On-Chip Communication

Nithin Michael, Yao Wang, G. Edward Suh, Ao Tang
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853, USA

I. Introduction

Network-on-chip (NoC) promises better scalability and power efficiency compared to traditional on-chip interconnects. But in order to fully exploit the benefits offered by the new paradigm, especially as the number of cores in the network increases, challenging resource management questions need to be addressed. Of particular interest and the subject of our study is the question of how to map applications to processors (network nodes) in a NoC so as to minimize the dynamic power consumption of the NoC.

II. Problem Setup

More specifically, we assume that each application consists of multiple parallel tasks, which communicate in a message passing fashion through an on-chip network. Each communication flow has a throughput requirement associated with it, which specifies how often a packet is sent between the corresponding source-destination pair in the network. The goal is to determine a mapping of application tasks to network nodes in a way that minimizes the communication energy while maintaining performance. We call this a "task mapping" problem.

We will use the following assumptions and notations. The NoC is assumed to have sufficient capacity to handle the applications that will be mapped to it assuming minimal routing. Minimal routing is assumed because it is commonly used and delivers good performance given enough capacity. Thus, we are left with the problem of mapping communicating tasks to processors to minimize dynamic power consumption. Suppose that there are V nodes in the NoC and that there are N tasks to be mapped for a given application. We also assume that $N \leq V$, as we are interested in the mapping aspect of the problem and not scheduling. Let A be the communication task graph of an application where $A(i,j)$ represents the communication rate from task i to task j. We will use $B(i,j)$ to represent the manhattan distance between node i and node j of the NoC. We want to minimize the communication cost, which given our assumptions, can be calculated as,

$$Cost = \sum_{i,j} A(i,j)B(p(i),p(j)) \qquad (1)$$

where $p(i), p(j)$ are the processors to which tasks i and j are mapped. The idea is that minimizing communication cost translates to minimizing dynamic power and leads to better energy-efficiency.

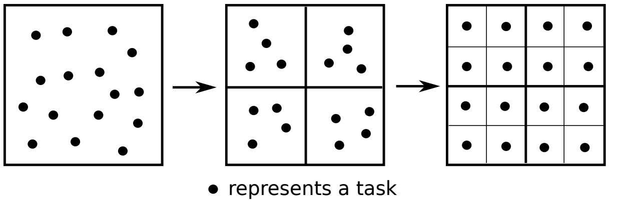


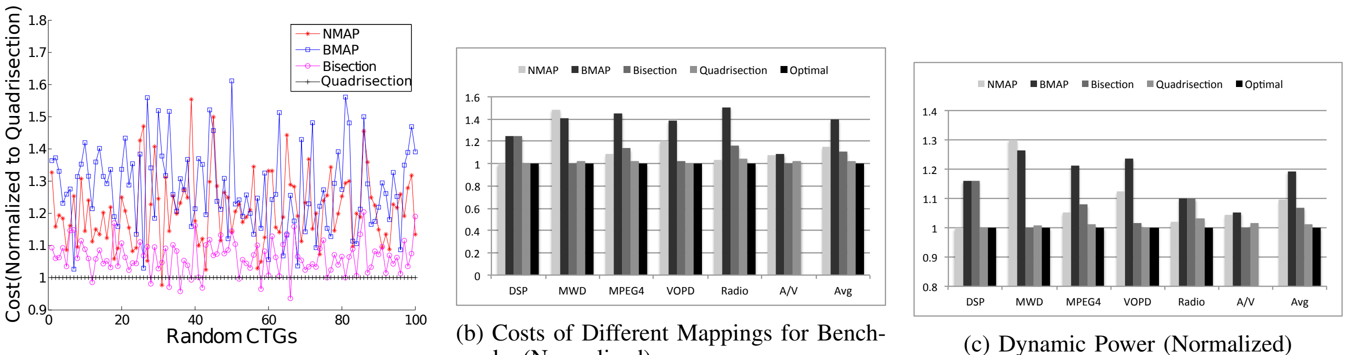
Figure 1: Quadrisection Method

Unfortunately, the task mapping problem is exactly the quadratic assignment problem which is not only NP-Hard but also difficult to approximate well in general [1]. However, for the special case of networks-on-chip, given the importance of the problem, multiple heuristic schemes have been developed for task mapping. We compare our work to three of these schemes NMAP [4], BMAP [5] and bisection [6] as we feel that they are representative of the ideas that are commonly used to tackle this problem in NoCs. We were motivated by the insight that mapping onto a NoC, particularly, one with a regular topology like the mesh topology, is a two-dimensional problem and should be approached as such. Unsurprisingly, a two-dimensional mapping approach had already been studied for VLSI layout [7]. While we adopt a similar approach, we also take advantage of the fact that we are mapping discrete tasks to individual processors to further refine the algorithm.

III. Quadrisection

As shown in Figure 1, the idea is to divide the tasks into four groups of the same size and map the groups to four smaller mesh networks (obtained by bisecting the original mesh network horizontally and vertically) in an attempt to minimize the communication crossing between the groups. We repeat this procedure recursively till the mapping problem becomes small enough to apply exhaustive search. We use the Fiduccia-Matthyses algorithm [3] to assign the tasks to each group. Given an initial assignment of the tasks to four groups, the Fiduccia-Matthyses algorithm moves one task at a time from one group to another in order to minimize the communication cost. To prevent infinite loops, once a task is moved, it is locked and cannot be moved

978-1-4673-6493-5/13/$31.00 ©2013 IEEE



(a) Total Communication Cost (Normalized to Quadrisection)

(b) Costs of Different Mappings for Benchmarks (Normalized)

(c) Dynamic Power (Normalized)

Figure 2: Evaluation results

again until the next iteration. At each iteration, for each task group that is further divided into sub-task groups, we take advantage of the discrete nature of our problem to perform exhaustive search in order to determine sub-task group to mesh assignments that minimize the communication cost to the other task groups. The high-level description of the core of the quadrisection algorithm is provided in Algorithm 1. In the algorithm, M_i refers to the coordinates that determine a mesh, S_i refers to a task group and $AssignLoc$ is used to assign sub-task groups to 'good' mesh locations with respect to external task groups.

Algorithm 1 Quadrisection(V, A)

```
Input: $V, A$

  $map \leftarrow [(0,0), \ldots, (0,0)]$                    ▷ Initialize $map$
  $map \leftarrow$RECURSIVEMAP$(M, S, A, map)$              ▷ Compute $map$
  procedure RECURSIVEMAP$(M, S, A, map)$
      if M is a $2 \times 2$ mesh then
          ▷ Do exhaustive search for $2 \times 2$ mesh & AssignLoc
          $map(S_1, S_2, S_3, S_4)$
      else
          $(M_1, M_2, M_3, M_4, S_1, S_2, S_3, S_4) \leftarrow FM(M, S, A)$
                                                  $\&AssignLoc$
                         ▷ Use Fiduccia-Matthyses to obtain sub-task groups
                         ▷ Use AssignLoc to determine mesh assignments
          For each $i = 1, 2, 3, 4$ do
              RECURSIVEMAP$(M_i, S_i, A, map)$
                         ▷ Do the mapping for each smaller mesh recursively
          end for
      end if
  end procedure
```

IV. Evaluation

The numerical evaluation was performed for synthetically generated random traffic patterns as well as for common benchmark patterns (Table I). As evident from Figure 2a, Quadrisection performed noticeably better than the other algorithms on almost all synthetic traffic patterns. Figure 2b shows that quadrisection outperformed again for the benchmark traffic patterns, coming very close to the optimal value for the cases for which we were able to calculate it. The benchmarks were also simulated using Darsim and the dynamic power evaluated using the power models from Orion 2. The results shown in Figure 2c were found to mirror our numerical predictions (Figure 2b). It should also be noted, particularly for online mapping tasks, that the runtime of these algorithms plays an important role in their selection. Promisingly, on average, quadrisection and bisection took on the order of 0.06 seconds like BMAP while NMAP took around 3.3 seconds. As can be seen from our evaluations, Quadrisection generates very good mappings quickly and should be a useful addition to the NoC designer's toolkit.

Table I: Benchmarks

Benchmark	# of tasks	# of flows	mesh size
DSP	6	8	4 × 4
MWD	12	13	4 × 4
MPEG4	12	26	4 × 4
VOPD	16	21	4 × 4
Ericsson Radio	15	26	4 × 4
A/V [2]	21	33	8 × 8

References

[1] E. Cela. *The Quadratic Assignment Problem: Theory and Algorithms*. Springer, 1998.

[2] V. Dumitriu and G. Khan. Throughput-Oriented NoC Topology Generation and Analysis for High Performance SoCs. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 17(10):1433 –1446, oct. 2009.

[3] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Design Automation Conference*, DAC '82, pages 175–181, Piscataway, NJ, USA, 1982. IEEE Press.

[4] S. Murali and G. D. Mitcheli. Bandwidth-constrained mapping of cores onto NoC architectures. In *Proceedings of DATE*, 2004.

[5] W.-T. Shen, C.-H. Chao, Y.-K. Lien, and A.-Y. A. Wu. A New Binomial Mapping and Optimization Algorithm for Reduced-Complexity Mesh-Based On-Chip Network. In *Proceedings of NOCS*, Princeton, United States, 2007. ACM.

[6] K. Srinivasan and K. S. Chatha. A Technique for Low Energy Mapping and Routing in Network-on-Chip Architectures. In *Proceedings of ISLPED*, San Diego, CA, 2005.

[7] P. Suaris and G. Kedem. An algorithm for quadrisection and its application to standard cell placement. *Circuits and Systems, IEEE Transactions on*, 35(3):294 –303, mar 1988.