

# Optimizing Data Intensive Flows for Network on Chip Mesh Networks

Junwei Zhang, *Member, IEEE*, Yang Liu, and Thomas Robertazzi, *Fellow, IEEE*<sup>‡</sup>

July 9, 2018

**Abstract**—Data flow analysis and optimization is considered for homogeneous rectangular mesh networks. Firstly, closed-form solutions are found for the equivalence computation [7] of divisible workload originating from a single data injection point in a homogeneous mesh network. We propose a flow matrix closed-form equation to present the equivalence, which allows a characterization of the nature of minimal time solution and a simple method to determine when and how much load to distribute to processors. We also propose a rigorous mathematical proof about the flow matrix optimal solution existence and that the solution is unique. Secondly, load originating at multiple data injection points is

considered. A reduced Manhattan distance Voronoi diagram algorithm (RMDVDA) is used to minimize the overall processing time of these workloads by taking advantage of a processor equivalence technique. In the first phase, a Voronoi Manhattan distance diagram is used to obtain a network cluster division. In the second phase, an efficient algorithm to obtain near-optimal load distribution among processors represented by equivalent processors is proposed. The algorithm minimizes the number of processors utilized. Experimental evaluation through simulations demonstrates that a load can be processed in about the same suboptimal time and yet save about 35% - 40% of processor resources. Further, the lower bound of intuitive and heuristic algorithms are also investigated. This work has important consequences for networks on chips where the demonstrated ability to reduce the number of processors needed for a computational problem has significant consequences for chip area and chip scalability.

\*Junwei Zhang, Ph.D. candidate of the Department of Applied Mathematics and Statistics, Stony Brook University, NY, 11794 USA. e-mail: junwei.zhang@stonybrook.edu

<sup>†</sup>Yang Liu, Ph.D. of the Department of Electrical and Computer Engineering, Stony Brook University, NY, 11794 USA. e-mail: yangliu.suny@gmail.com

<sup>‡</sup>Thomas Robertazzi, professor of the Department of Electrical and Computer Engineering, Stony Brook University, NY, 11794 USA. e-mail: Thomas.Robertazzi@stonybrook.edu

**Index Terms**—Divisible Load Theory, Voronoi Diagram, Multi-source, Network on Chip (NOC), In-

**terconnection Network, Mesh, Data Intensive Load**

to a variety of switching/scheduling protocols besides those directly covered in this paper.

## I. BACKGROUND

Networks on chips (NOC) represent the smallest networks that have been implemented to date [6]. A popular choice for the interconnection network on such networks on chips is the rectangular mesh. It is straightforward to implement and is a natural choice for a planar chip layout. Data to be processed can be inserted into the chip at one or more so-called “injection points”, that is node(s) in the mesh that forward the data to other nodes. Beyond NOCs, injecting data into a parallel processor’s interconnection network has been done for some time, notably in IBM’s Bluegene machines [5]. In this paper it is sought to determine, for a given set of injection points on a heterogeneous rectangular mesh, how, optimally or near-optimally, to assign load to different processors/links in a known timed pattern so as to process a load of data in a minimal amount of time (i.e. minimize makespan). In this paper we succeed in presenting an optimal technique for single injection points in homogeneous meshes that involves no more complexity than linear equation solution. For multiple injection points we present algorithms that produce near optimal solutions using Voronoi diagrams. The methodology presented here can be applied

## II. APPROACH

### A. *Divisible Load Theory*

Crucial to our success in the single and multiple injection point cases, is the use of divisible load scheduling theory [1,2]. Developed over the past few decades, it assumes load is a continuous variable that can be arbitrarily partitioned among processors and links in a network. Use is made of the divisible load scheduling’s optimality principle [1] [8], which says makespan is minimized when one forces all processors to stop at the same time (intuitively otherwise one could transfer load from busy to idle processors to achieve a better solution). This leads to a series of chained linear flow and processing equations that can be solved by linear equation techniques, often yielding recursive and even closed form solutions for quantities such as makespan and speedup.

### B. *Voronoi Diagrams*

We utilize Voronoi diagrams in conjunction with divisible load scheduling in this work. In mathematics, a Voronoi diagram [2] is a partitioning of a plane into regions based on distance to points in a specific subset of the plane. For each seed there is a corresponding region consisting of all points closer to that

seed than to any other. These regions are called Voronoi cells.

### *C. Virtual Cut-through and Equivalence Computation*

In this paper, the use of virtual cut-through switching [4] is investigated. This is one of many switching protocols that our methodology applies to. In the virtual cut-through environment, a node can begin relaying the first part of a message (packet) along a transmission path as soon as it starts to arrive at the node, that is, it doesn't have to wait to receive the entire message before it can begin forwarding the message. Equivalence computation [7] is a technique, which consists of representing a network of processors exactly as a single equivalent processor of known processing speed.

### III. SINGLE INJECTION POINT CASE

First, an equivalent processor (and processor load fractions and speedup) is found for a  $2 \times 2$  homogeneous mesh network, which can be generalized to a homogeneous  $2 \times n$  mesh network. After that, we analyze a more general case homogeneous  $m \times n$  mesh network and obtain a general closed-form matrix representation yielding a processor with equivalent processor speed and speedup and processor/link load allocation. Finally, a key methodology is given to address this type of question. In addition, different single data injection point

positions, such as the corner, boundary and inner grid are also discussed. In addition, a rigorous mathematical proof about the flow matrix solution's existence and uniqueness is presented.

### IV. MULTIPLE INJECTION POINT CASE

First, data injection points consisting of a connected subgraph in a mesh network are considered. An equivalence processor scheduling algorithm (EPSA), which considers a "big" source comprised of individual connected sources is proposed. Then, the flow matrix is utilized to calculate processors' data fractions, individually. Second, if the data injection points don't connect with each other, an intuitive Manhattan distance Voronoi diagram algorithm (MDVDA) is proposed to solve this problem, which obtains a partitioning of the mesh network and each cell's corresponding data injection point's workload. Third, it is found that the makespan depends on the bottleneck makespan. In other words, if other Voronoi cells consist of more processors than the bottleneck cell has, one can improve the makespan by addressing the bottleneck region. A heuristic algorithm reduced Manhattan distance Voronoi diagram algorithm (RMDVDA) is presented. This algorithm importantly reduces the number of processors needed for processing with little change in the makespan. As an example, we choose a  $50 \times 50$  mesh network and 10 data

injection locations. After 1000 rounds of random sampling, it is found that this algorithm can solve a divisible load in about the same suboptimal time as MDVDA, yet save about 35%-40% of the number of required processors. In addition, a rigorous mathematical proof of a lower bound of our three algorithms is given.

## V. SIGNIFICANCE OF WORK

In sum, in this work, a flow matrix quantitative model, which gives when and how to deploy the data fraction to each processor in a homogeneous mesh, individually, is proposed. It is proven that solution exists and the flow matrix solution is unique. The complexity of the technique is no more than that of linear equation solution complexity. Further, three algorithms, EPSA, MDVDA and RMDVDA, are proposed to solve the multi-source assignment problem depending on the data injection's connection position. We give a rigorous proof about the efficiency of the algorithms. This work has relevance to mesh interconnection networks used in parallel processing in general and to meshes used in Networks on Chips in particular. The ability to reduce the number of processors used in a divisible style computation on a NOC has important consequences for required chip area and chip scalability. After 1000 rounds of simulation, savings of about 35%-40% processor resources with ap-

proximately the same solution makespan were found.

## VI. RELATED WORK

In the context of multiple injection point models, this paper represents Jia [3] proposes a genetic algorithm, which utilize a novel Graph Partitioning (GP) scheme to partition the network such that each source in the network gains a portion of network resources and then these sources cooperate to process their loads.

## VII. CONCLUSION

This paper represents an advance in our ability to analytically and numerically evaluate the flow of data originating at single/multiple injection points in a rectangular mesh network. The methodology is applicable to a variety of switching and load distribution protocols. An important application is improving chip area and chip scalability for chips processing divisible style loads.

## REFERENCES

- [1] Veeravalli Bharadwaj et al. *Scheduling divisible loads in parallel and distributed systems*. Vol. 8. John Wiley & Sons, 1996.
- [2] Steven Fortune. "Voronoi diagrams and Delaunay triangulations". In: *Computing in Euclidean geometry*. World Scientific, 1995, pp. 225–265.

- [3] Jingxi Jia, Bharadwaj Veeravalli, and Jon Weissman. “Scheduling multisource divisible loads on arbitrary networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 21.4 (2010), pp. 520–531.
- [4] Parviz Kermani and Leonard Kleinrock. “Virtual cut-through: A new computer communication switching technique”. In: *Computer Networks (1976)* 3.4 (1979), pp. 267–286.
- [5] Elie Krevat, José G Castaños, and José E Moreira. “Job scheduling for the Blue-Gene/L system”. In: *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer. 2002, pp. 38–54.
- [6] Thomas G Robertazzi. *introduction to computer networking*. Springer Science & Business Media, 2017.
- [7] Thomas G Robertazzi. “Processor equivalence for daisy chain load sharing processors”. In: *IEEE Transactions on Aerospace and Electronic Systems* 29.4 (1993), pp. 1216–1221.
- [8] Jeeho Sohn and Thomas G Robertazzi. *Optimal load sharing for a divisible job on a bus network*. Tech. rep. Stony Brook, NY: State University of New York at Stony Brook, College of Engineering., 1992.