Optimal Data Intensive Flows for Network on Chip Mesh Networks (what is your opinion?)

Junwei Zhang*, Yang Liu**, Esther Arkin * and Thomas Robertazzi**,

*Dept. of Applied Mathematics and Statistics,

**Dept. of Electrical and Computer Engineering,

Stony Brook University,

Stony Brook NY 11794

Junwei.Zhang@stonybrook.edu

ABSTRACT

This thesis considers two problems. One problem is closed-form solutions for equivalence computation [11] of divisible workload in a mesh networks and the other problem is scheduling divisible workloads from multiple sources in mesh networks of processors. 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 for processors. In addition, we also propose a rigorous mathematics proof about the flow matrix optimal solution existence and unique. Also, we propose the use of a reduced Manhattan distance Voronoi diagram algorithm (RMDVDA) to minimize the overall processing time of these workloads by taking advantage of the processor equivalence technique. The user case studies with 10 sources of workloads are presented to illustrate the general approach for multiple sources of workloads. In the first phase, a Voronoi Manhattan distance diagram is used to obtain a network cluster division. In the second phase, we propose an efficient algorithm to obtain near-optimal load distribution among processors represented by equivalent processors. The algorithm minimizes the number of processors utilized. Experimental evaluation through simulations demonstrates that a task can be finished in the same suboptimal time and yet save

about 35% - 40% of processor resources. Further, the lower band of intuitive and heuristic algorithm is also investigated.

1.0 Background

Networks on chips (NOC) represent the smallest networks that have been implemented to date [10]. 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 [8].

In this paper it is sought to determine, for a given set of injection points how, optimally or near-optimally, to assign load to different processors 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 to a variety of switching/scheduling protocols besides those directly covered in this paper.

2.0 Approach

2.1 Divisible Load Theory

Crucial to our success in the single and multiple injection point cases, is the use of divisible load scheduling theory [1]. 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 [2], which say 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.

2.2 Voronoi Diagrams

In the context of multiple injection point models, this paper represents Jia [6] 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. We utilize the Voronoi diagrams in conjunction with divisible load scheduling for a significant applied problem.

In mathematics, a Voronoi diagram [4] 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.

2.3 Virtual Cut-through and Equivalence Computation

In this paper, we investigate the virtual cut-through switching [7]. 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 [11] is a technique, which consists of combining a cluster of processors as one whole equivalent processor to process a unit 1 workload..

3.0 Single Injection Point Case

First, we consider about the $2 * 2$ mesh network, which can be generalized to a $2 * n$ mesh network. After, we analyze a more general case $m*n$ mesh network and obtain a general closed-form matrix presentation. Finally, we give a key methodology to address this type of question. In addition, different single data injection position, such as the corner, boundary and inner grid are also discussed.

In addition, we also propose a rigorous mathematics proof about the flow matrix solution's existence and unique.

4.0 Multiple Injection Point Case

First, we consider about the data injections consist of a connected subgraph of mesh network. We propose an equivalence processor scheduling algorithm (EPSA), which consider the connected subgraph data injection consists of "big" source. Then, we utilize the flow matrix to calculate processors' data fraction, individually.

Second, if the data injections don't connect with each other, we propose an intuitive Manhattan distance Voronoi diagram Algorithm (MDVDA) to tackle this problem, which obtains a partitioning of the mesh network and each cell tackle corresponding data injection's workload.

Third, after investigation, we find the makespan depends on the bottleneck makespan. In other words, if other divisions own more processors than the bottleneck cell, it does not help to minimize the makespan. We propose a heuristic algorithm reduced Manhattan distance Voronoi diagram algorithm (RMDVDA). In user case, we choose 50*50 mesh network and 10 data injections, after 1000 rounds random sampling, we find this algorithm can utilize the same suboptimal time as MDVDA, yet save about 35%-40% processor resources.

In addition, we also give a rigorous mathematics proof of a lower band of our three algorithms.

5.0 Significance of Work

In sum, in this work, we propose a flow matrix quantitive model, which gives when and how to deploy the data fraction to each processor, individually. We also prove the solution exists and unique of the flow matrix solution.

Also, we propose three algorithms, EPSA, MDVDA and RMDVDA, which tackle the multi-source assignment problem depending on the data injection's connection position. Further, we also give a rigorous proof about the efficiency of your algorithm.

In addition, after 1000 rounds simulation, we save about 35%-40% processor resources within the same suboptimal time.

~~6.0 Related Work (maybe) – mention Indian paper I already mention in Approach~~

~~7.0 Conclusion~~

REFERENCES

[1] Veeravalli Bharadwaj, Debasish Ghose, and Thomas G Robertazzi. "Divisible load theory: A new paradigm for load scheduling in distributed systems". In: *Cluster Computing* 6.1 (2003), pp. 7–17.

[2] Veeravalli Bharadwaj et al. *Scheduling divisible loads in parallel and distributed systems*. Vol. 8. John Wiley & Sons, 1996.

[3] Steven Fortune. “A sweepline algorithm for Voronoi diagrams”. In: *Algorithmica* 2.1-4 (1987), p. 153.

[4] Steven Fortune. “Voronoi diagrams and Delaunay triangulations”. In: *Computing in Euclidean geometry*. World Scientific, 1995, pp. 225–265.

[5] Jui Tsun Hung and Thomas G Robertazzi. “Switching in sequential tree networks”. In: *IEEE Transactions on Aerospace and Electronic Systems* 40.3 (2004), pp. 968–982.

[6] 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.

[7] Parviz Kermani and Leonard Kleinrock. “Virtual cutthrough: A new computer communication switching technique”. In: *Computer Networks (1976)* 3.4 (1979), pp. 267–286.

[8] Elie Krevat, Jose G Casta ´ nos, and Jos ˜ e E Moreira. ´ “Job scheduling for the BlueGene/L system”. In: *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer. 2002, pp. 38–54.

[9] Xinxin Liu, Han Zhao, and Xiaolin Li. *Scheduling Divisible Workloads from Multiple Sources in Linear Daisy Chain Networks*.

[10] Thomas G Robertazzi. *introduction to computer networking*. Springer Science & Business Media, 2017.

[11] 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.

[12] 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