Contention-Aware Mapping and Scheduling Optimization for NoC-Based MPSoCs (Student Abstract)

Yupeng Zhou,[1] **Rongjie Yan,**[2,*] **Anyu Cai,**[3] **Yige Yan,**[1] **Minghao Yin**[1,*]
[1]School of Information Science and Technology, Northeast Normal University, China
[2]Institute of Software Chinese Academy of Sciences, China
[3]School of Computer and Information Technology, Beijing Jiaotong University, China
{zhouyp605, yanyg270, ymh}@nenu.edu.cn, yrj@ios.ac.cn, anyucai@outlook.com

Abstract

We consider spacial and temporal aspects of communication to avoid contention in Network-on-Chip (NoC) architectures. A constraint model is constructed such that the design concerns can be evaluated, and an efficient evolutionary algorithm with various heuristics is proposed to search for better solutions. Experimentations from random benchmarks demonstrate the efficiency of our method in multi-objective optimization and the effectiveness of our techniques in avoiding network contention.

Introduction

Network-on-Chip (NoC) has emerged as an alternative interconnecting paradigm in the state-of-the-art multi-core architectures. An important issue in the NoC paradigm is how to map and schedule tasks of an application to processing elements (PEs), such that the total energy consumption is minimized, and system performance can be optimized (Sahu and Chattopadhyay 2013). Within this architecture, communication is the main concern in the optimization process. To reduce the cost, tightly coupled tasks will be closely allocated, which may increase the possibility of communication *contention* for frequent data transfer over same paths. The increase in contention may incur long latency from network congestion thus leading to high energy consumption and poor system performance (Yang et al. 2016; Chou and Marculescu 2008).

With contention awareness in the design of NoCs, performance and energy consumption optimization needs to consider how to map tasks of an application to available PEs and tiles (the basic building block which can accommodate one or more PEs), and how to schedule tasks on same PEs to avoid contention. However, path-based contention minimization may reduce the degree of parallel execution between various tasks (Chou and Marculescu 2008). Introducing additional latency to avoid overlapped communication in scheduling may degrade system performance. Therefore, reducing contention with less influence on performance and energy requires wary mapping and scheduling strategies, which is a challenge for NoCs.

In this paper, we consider time-triggered static scheduling for NoC-based MPSoCs, where both mapping and scheduling cannot be modified at runtime. To optimize performance and energy consumption with the consideration of potential contentions in communication, we firstly construct formulations for mapping and scheduling constraints and objectives to be optimized. Meanwhile, we design three encodings for MILP, CP, SMT solving respectively. To deal with large-scale applications, we further integrate various heuristics and a problem-specific local search into a multi-objective evolutionary algorithm.

Constraint formulation for NoC-based design

We provide a flexible constraint formulation for NoC-based mapping and scheduling in the format of logical formulas, which can be translated to other forms such as constraint or mixed linear programming formulas for various solvers.

$$m_{ik} \rightarrow \neg(\bigvee_{k' \neq k} m_{ik'}), \quad \ell_{ik} \rightarrow \neg(\bigvee_{k' \neq k} \ell_{ik'}) \tag{1}$$

$$(\sum_{i=1}^{|T|} m_{ik}) \leq \omega_k \tag{2}$$

$$d_{ij} \wedge m_{ik} \wedge m_{jk'} \wedge (k \neq k') \rightarrow o_{ij}, \;\; o_{ij} \rightarrow d_{ij} \tag{3}$$

$$(\gamma_{k_1 k_2 k_3 k_4} > 0) \wedge \ell_{ik_1} \wedge \ell_{jk_2} \wedge \ell_{lk_3} \wedge \ell_{rk_4} \wedge o_{ij} \wedge o_{lr} \rightarrow cf_{ijlr} \tag{4}$$

$$f_i^u = s_i^u + a_i / (\sum_{k=1}^{|P|} m_{ik} \cdot \rho_k) \tag{5}$$

$$\hat{f}_{ij}^u \geq \hat{s}_{ij}^u + o_{ij} \cdot (c_{ij} \cdot \tau \cdot D_{ij} / bw + \tau' \cdot (D_{ij} + 1)) \tag{6}$$

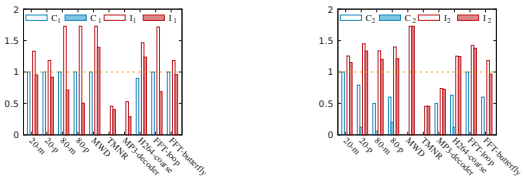
$$\bigvee_{j=1}^{|T|} d_{ji} \rightarrow \hat{f}_{ji}^u \leq s_i^u, \quad f_i^u \leq \hat{s}_{ij}^u \tag{7}$$

$$n_{ij} \rightarrow f_i^u \leq s_j^v, \text{for } u \leq v \tag{8}$$

$$m_{ik} \wedge m_{jk} \rightarrow f_j^u \leq s_i^v, \text{for } u < v \tag{9}$$

$$m_{ik} \wedge m_{jk} \rightarrow s_j^u \geq f_i^u \vee s_i^u \geq f_j^u \tag{10}$$

*Corresponding author



(a) Local search (b) Initialization

Figure 1: Comparison between heuristics

$$cf_{ijlr} \wedge abs(f_i^u - f_l^u) \leq \xi \rightarrow \hat{s}_{ij}^u \geq \hat{f}_{lr}^u \vee \hat{s}_{lr}^u \geq \hat{f}_{ij}^u \quad (11)$$

The makespan of executing an application within N iterations and the energy costs of computation and communication are evaluated as follows.

$$\mathcal{M} = \max\{f_i^N \mid t_i \in T\} \quad (12)$$

$$E_p = \sum_{p_k \in P'} (N \cdot \mathcal{E}_{d_k} \cdot \mathbb{T}_k + \mathcal{E}_{i_k} \cdot (\mathcal{M} - N \cdot \mathbb{T}_k)) \quad (13)$$

$$E_m = \sum_{p_k, p_{k'} \in P'} \sum_{i,j=1}^{|T|} c_{ij} o_{ij} m_{ik} m_{jk'} (\varepsilon D_{kk'} + \varepsilon'(D_{kk'} + 1)) \quad (14)$$

Suppose two paths are from PE p_{k_1} to p_{k_2}, and p_{k_3} to p_{k_4}, respectively. Their contention probability is

$$\wp_c(k_1, k_2, k_3, k_4) = \gamma_{k_1 k_2 k_3 k_4} / (D_{k_1 k_2} \cdot D_{k_3 k_4}) \quad (15)$$

$$\mathcal{P}_{i_1 i_2} = \sum_{j_1 \in \mathcal{S}_{i_1}, j_2 \in \mathcal{S}_{i_2}} o_{i_1 j_1} \cdot o_{i_2 j_2} \cdot \wp_c(k_{i_1}, k_{j_1}, k_{i_2}, k_{j_2}) \quad (16)$$

Therefore, the sum of contention probabilities for a given mapping strategy is $\mathcal{P}_c = \sum_{i,j=1}^{|T|} \mathcal{P}_{ij}$.

To avoid high local contention probability, we apply the average contention probability to evaluate the quality of the allocation strategy, which is denoted by Equation 17.

$$\bar{\mathcal{P}}_c = \sum_{i,j=1}^{|T|} abs(\mathcal{P}_{ij} - \mathcal{P}_c / |T|) \quad (17)$$

Based on the discussion above, we have three objectives to be optimized: makespan, energy cost, and contention:

$$minimize(\mathcal{M}),\ minimize(E_p + E_m), minimize(\bar{\mathcal{P}}_c) \quad (18)$$

The multi-objective hybrid algorithm

We integrate a Pareto local search method into the evolutionary framework together with objective-related heuristic extensions. The algorithm consists of three optimization stages. The first is a task-classification process, including task clustering (TC) and cluster refinement (CR), to group tightly coupled tasks into one PE. Then, with the heuristic of spiral mapping (SM), a PE type and location mapping stage is applied to optimize the makespan and average contention probability. The two stages are integrated into initialization. In the evolution process with mutation and crossover operators, we look for a better scheduling sequence for tasks of every PE, to minimize the makespan and satisfy the response time requirement between certain tasks. To sum up, we have provided four methods to tackle the contention-aware NoCs optimization, including MILP solving, CP solving, SMT solving and heuristic solving.

Table 1: Comparison between MOHA and various methods

Case	$\|T\|$	$\|E\|$	MOHA	NSGAII	CPLEX(MILP)		CPLEX(CP)		Z3	
			sol	sol	sol	time	sol	time	sol	time
5-m	5	4	3	2(=)+2(≻)	2(=)	687.63	2(=)	53.62	3(=)	3.68
5-p	5	4	3	3(=)	2(=)	829.39	2(=)	49.22	3(=)	6.49
7-m	7	6	2	2(=)	2(=)	-	2(=)	158.96	2(=)	9.14
7-p	7	6	2	2(=)	2(=)	-	2(=)	247.59	2(=)	19.85
8-m	8	7	2	2(=)	1(=)+1(≻)	-	1(=)+1(≻)	287.50	2(=)	12.92
8-p	8	7	2	2(=)	2(=)	-	2(=)	316.83	2(=)	23.13
10-m	10	9	1	1(=)	1(=)	-	1(=)	-	1(=)	93.45

Experimental results

The effectiveness of the model and the efficiency of proposed algorithm (MOHA) are evaluated with randomly generated cases from TGFF tool. We compare the performance among MOHA, exact methods (MILP solver and CP solver of CPLEX, Z3) and NSGAII for tested cases.

In Table 1, the MOHA column presents the number of non-dominated solutions. The solution in CPLEX (or NSGAII, Z3) column shows the number of solutions and the relation between the generated solutions with MOHA, where $\succ$ means that solutions of CPLEX (or NSGAII) are dominated by those of MOHA, and $=$ means that solutions of MOHA and CPLEX (or Z3, NSGAII) are the same. In the first case, though NSGAII can find four solutions, two of them are dominated by those of MOHA. Z3 can always provide all the solutions, for the constraints can be encoded as logic formulas and the optimization process is fast for small-scale cases. Compared with the MILP solver, CP solver also achieved better results from the logic constraint reasoning.

To check the effectiveness of the heuristics, we present the comparison from Set Coverage (blue columns) and hyper volume indicator I_H (red columns) to reflect both dominance degree and distribution of different pareto fronts. Note that hollow columns in Fig. 1 represent the proposed algorithm, while the solid ones stand for the comparing algorithms. It reveals that MOHA can find better solutions with the local search and initialization process respectively, which indicates the validity of the heuristics.

References

Chou, C.-L., and Marculescu, R. 2008. Contention-aware application mapping for network-on-chip communication architectures. In *ICCD*, 164–169.

Sahu, P. K., and Chattopadhyay, S. 2013. A survey on application mapping strategies for network-on-chip design. *JSA* 59:60–76.

Yang, L.; Liu, W.; Jiang, W.; Li, M.; Yi, J.; and Sha, E. H.-M. 2016. Application mapping and scheduling for network-on-chip-based multiprocessor system-on-chip with fine-grain communication optimization. *TVLSI* 24(10):3027–3040.