Computers and Electrical Engineering 68 (2018) 581–602

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Hotspot-aware task-resource co-allocation for heterogeneous many-core networks-on-chip☆

Md Farhadur Reza*, Dan Zhao[1], Hongyi Wu[2], Magdy Bayoumi[3]

The Center for Advanced Computer Studies, University of Louisiana at Lafayette, 301 East Lewis Street, Lafayette, LA 70503, USA

ARTICLE INFO

Keywords:
Task-resource co-allocation
Network-on-Chip (NoC)
Many-Core NoC
Heterogeneous
Computation and communication
Peak load
Energy hotspots
Cost budget
Linear programming (LP)
Heuristic

ABSTRACT

To fully exploit the massive parallelism of many-core on a chip, this work tackles the problem of mapping large-scale applications onto heterogeneous networks-on-chip (NoCs) while minimizing hotspots. A task-resource co-optimization framework is proposed which configures the on-chip communication infrastructure and maps the applications simultaneously and coherently, aiming to minimize the peak energy under the constraints of computation power, communication capacity, and total cost budget of on-chip resources. The problem is first formulated into a linear programming model to search for optimal solution. A heuristic is further developed for fast design space exploration at design-time and run-time in large-scale NoCs. Extensive simulations are carried out under real-world benchmarks and randomly generated task graphs to demonstrate the effectiveness and efficiency of the proposed schemes. Real system simulations show the significant improvement (30–200%) in NoCs latency and throughput compared to the state-of-the-art minimum-path approach because of the diminishing hotspots and balanced load distribution.

1. Introduction

Multicore chip design has been increasingly deployed in embedded computing systems ranging from small mobile devices (e.g., Nvidia Tegra 3) to large telecommunication servers (such as ARMv8-A) in near future to meet ever-increasing computational demands under a stringent energy budget. A single chip with many-cores can replace a traditional server or a rack of servers in a data center. Many-core on-chip systems have better power efficiency, interconnection, and latency compared to traditional local area network (LAN) based systems [1]. Industries and academicians are working to integrate many-cores on a chip that can be comparable with the big data-center solution. For example, a team of researchers at the University of California Davis has implemented (with the help of IBM) a 1000-core chip [2], which has a maximum computation rate of 1.78 trillion instructions per second and contains 621 million transistors. The companies, who have their own data-center (e.g., Google and Facebook), are working with the chip companies (e.g., Intel) to get a many-core chip solution which can replace their big data-center solutions. Qualcomm has unveiled a 48-core 64-bit ARMv8 processor for servers in data centers [3]. Other companies (e.g., AMD, Amazon) have also taken initiatives towards many-core on-chip systems.

☆ Reviews processed and approved for publication by Editor-in-Chief.

* Corresponding author. Current affiliation: Department of Electrical & Computer Engineering, The George Washington University, 800 22nd Street NW, 5000 Science & Engineering Hall, Washington, DC 20052, USA.

E-mail addresses: mxr7945@louisiana.edu, farhadur_reza@gwu.edu (M.F. Reza), dzhao@louisiana.edu, zhao@cs.odu.edu (D. Zhao), wu@louisiana.edu, h1wu@odu.edu (H. Wu), mab0778@louisiana.edu (M. Bayoumi).

[1] Current affiliation: Department of Computer Science, Old Dominion University, 3300 Engineering & Computational Sciences Building, Norfolk, VA 23529, USA.

[2] Current affiliation: Department of Electrical & Computer Engineering, Old Dominion University, 231 Kaufman Hall, Norfolk, VA 23529, USA.

[3] Current affiliation: Department of Electrical & Computer Engineering, University of Louisiana at Lafayette, 131 Rex Street, Lafayette, LA 70504, USA.

https://doi.org/10.1016/j.compeleceng.2018.04.019
Received 12 May 2017; Received in revised form 24 April 2018; Accepted 25 April 2018

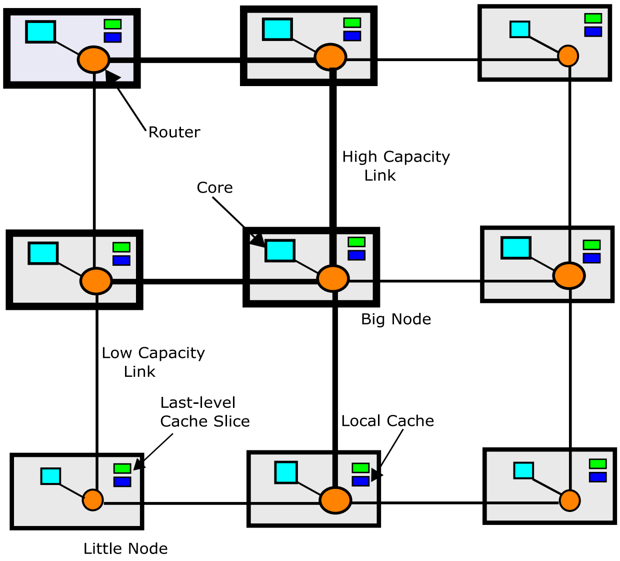

Fig. 1. Heterogeneous many-core architecture on NoC.

On-chip systems are moving towards the integration of tens or hundreds of heterogeneous cores (e.g., central processing unit (CPU), graphics processing unit (GPU), digital signal processing (DSP), accelerator) on a single chip to accomplish emerging applications through low-power highly parallel computing. The heterogeneous many-core architecture offers a feasible approach that embraces high-performance "Big" cores (e.g., CPUs) for computation intensive applications and low-power "Little" cores (e.g., GPUs and DSPs) for majority workloads execution. As the number of cores increases, network-on-chip (NoC) has been adopted as a viable solution to manage complex interconnection networks. NoC offers several important benefits over the traditional bus in terms of scalability, parallelism, throughput, and power efficiency [4]. For example, a many-core architecture on NoC is illustrated in Fig. 1 where heterogeneous processor tiles are placed in a 2D grid. Each tile contains a processor (with distinct computation power), its local cache, a slice of the shared last-level cache, and a router (configured with different communication capacity) for data and control transmission between the tiles via varying bandwidth links.

1.1. Task allocation in many-core NoCs

A many-core server may run multiple applications simultaneously. To fully utilize the computation power (or computation capacity) of the cores, an application can be partitioned into a number of tasks to be simultaneously processed on different processor tiles. Meanwhile, multiple tasks with diverse computation workloads from different applications may share the same tile under the restricted on-chip resource budget, as illustrated in Fig. 2. While task partitioning is beyond the scope of this work, we assume each application has been appropriately partitioned into a set of tasks that can be executed concurrently. Though tasks of the same application are contiguously mapped in Fig. 2, tasks of the same application can be remotely mapped in real scenario, which will be discussed soon. Data are transferred between communication dependent tasks via NoC to fulfill computation of the applications. NoC, as a communication network, also consumes a significant percentage of total chip power [5]. For example, on-chip network consumes 36% of total chip power in 16-tile MIT Raw [5]. The NoC power consumption has been increasing with the increase in core integration on the chip. Heterogeneous traffic loads are common in NoC, e.g., higher uniform traffic loads are exhibited in the center of

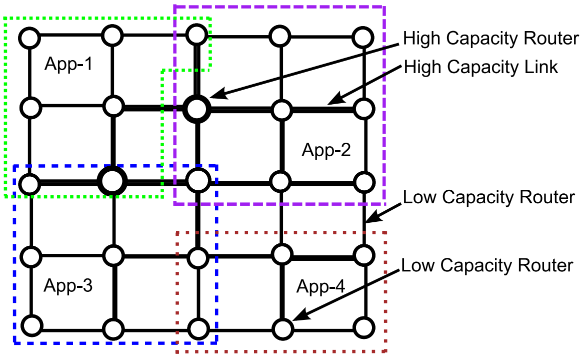

Fig. 2. Application partitioning and mapping in a many-core NoC.