Memory-Aware Mapping and Scheduling of Tasks and Communications on Many-Core SoC

Jinho LEE and Kiyoung Choi
Design Automation Lab., Seoul National University, KOREA
icarosj@dal.snu.ac.kr, kchoi@snu.ac.kr

Abstract

There have been many researches on mapping and scheduling tasks on a many-core SoC. However, none of them consider alternative ways of implementing communications, which can affect the entire system's performance, energy consumption and local memory usage.

This poster presents an approach to automatic task mapping, scheduling, and communication routing on a many-core SoC. In addition to mapping and scheduling the tasks, it considers the trade-offs between two different communication methods - message passing and shared memory - for the communication routing in order to optimize the energy consumption or performance.

Considering that the overhead of shared memory communication is relatively large, the proposed approach adds an AMP (active memory processor) to the shared memory tile and moves automatically chosen code blocks containing frequent memory operations from a processor core to the AMP, which significantly reduces the traffic between the core tile and the memory tile.

To solve the entire optimization problem, the proposed approach uses simulated annealing as well as optimization heuristics. It performs scheduling of the tasks even with backward dependencies using the iterative modulo scheduling technique. Experiments with random task graphs as well as a set of real applications show the effectiveness of the proposed approach.
Starting with the overview, it appears that the document focuses on memory-aware mapping and scheduling of tasks and communications on many-core SoC. The authors, Jinho LEE and Kyoung Choi, are associated with the Design Automation Lab, Seoul National University, KOREA, with email addresses icleo@dal.snu.ac.kr, kchoi@snun.ac.kr.

The main diagram on the page illustrates the use of message passing on a SM machine. It highlights the following points:

- **Overall Procedure**: The process includes obtaining mapping and communication methods, optimizing, coarsely-grained modulo scheduling, evaluating, and replacing solutions. The flow is marked as A. Obtain mapping and communication methods, B. Optimize, C. Coarse-grained modulo scheduling, D. Evaluate, and E. Replace solutions.

- **Coarse Grained Modulo scheduling**: Original modulo scheduling attempts to schedule tasks at specific intervals, which may fail due to non-minimal paths and SM delays. Coarse-grained scheduling tries to schedule tasks at the end of conflicting tasks, aiming for faster execution with minimal loss of quality.

- **Experimental Result**: The graph shows energy consumption and initiation interval comparisons between different methods, including AMP (Active Memory Processor) and Hybrid+AMP.

- **Communication Fix Optimization**: This section focuses on optimization techniques such as greedy heuristics and gain calculations based on distance and communication volume in relation to SPM usage.

- **Backgrounds**: The background diagram provides a network perspective, with nodes and edges indicating mapping and scheduling processes.

The content also includes sections on baseline and proposed methods, with graphical representations showing reduced execution time and improved energy efficiency compared to traditional methods.