PAXES – Parallelism Extraction for Embedded Systems Three Approaches – One Tool

Daniel Cordes, Peter Marwedel
TU Dortmund University
Dortmund, Germany
firstname.lastname@tu-dortmund.de

Abstract

Nowadays, complexity and performance requirements of embedded software are continuously increasing, making Multiprocessor System-on-Chip (MPSoC) architectures more and more important in the domain of embedded systems. Using multiple cores in one system reduces problems concerning energy consumption and heat dissipation. In addition, performance can also be dramatically increased. Nevertheless, these benefits do not come for free. Porting existing, mostly sequentially written, applications to MPSoCs requires extracting efficient parallelism to utilize all available cores. Therefore, we developed the PAXES (Parallelization Extraction for Embedded Systems) tool which is tailored towards the special requirements of resource restricted embedded systems. Up to now, three approaches are combined in this tool. The first one extracts coarse-grained task-level parallelism by deploying an integer linear programming (ILP) based approach. The second approach also employs ILPs to extract more fine-grained pipeline parallelism from loops of the applications. Both approaches can also be combined to extract both kinds of parallelism. However, in many cases it is not sufficient to only optimize the execution time of the application. Therefore, our third approach deploys a multi-objective aware extraction of task-level parallelism. By using this approach, the designer can choose an optimal combination between execution time reduction, energy consumption and communication load.

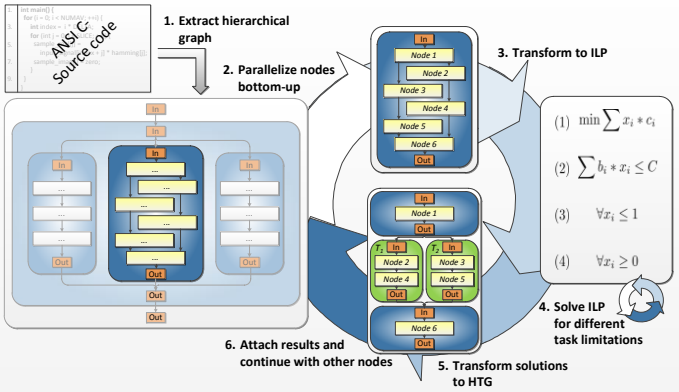
technische universität dortmund

PAXES – Parallelism Extraction for Embedded Systems
Three Approaches – One Tool

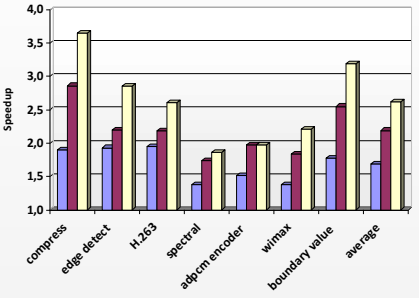
Motivation

- Resource-Restricted embedded systems have many limitations, compared to high-performance / desktop computing
- Consider these restrictions in the parallelization process
- Balance created tasks to extract really efficient parallelism
- Make use of high-level cost models to know where parallel execution really increases the performance
- Take care that benefit of parallelization is not eliminated by task-creation and communication overhead

ILP-based Extraction of Coarse-Grained Task-Level Parallelism

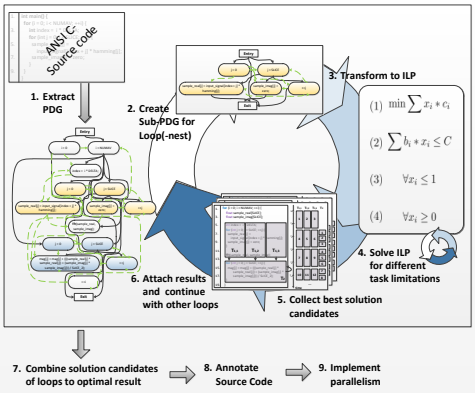


- Approach extracts very coarse-grained task-level parallelism (e.g., two parallel function calls) from sequential ANSI-C applications
- Extraction algorithm is based on integer linear programming (ILP)
- Clear mathematical problem description
- No approximation in extraction step
- Hierarchical Task Graph used as intermediate representation
- ILPs can be solved very efficiently (< 1 second in most cases)
- Speedup of up to 1.9x, 2.9x and 3.7x (on 2, 3, and 4 core architectures)

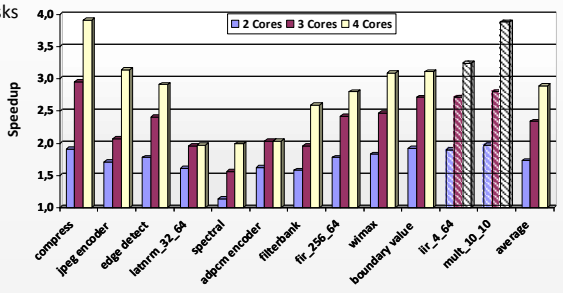


[1] Daniel Cordes, Peter Marwedel, and Arindam Mallik. *Automatic Parallelization of Embedded Software Using Hierarchical Task Graphs and Integer Linear Programming*. In Proc. of CODES+ISSS 2010.

ILP-based Extraction of Pipeline-Parallelism

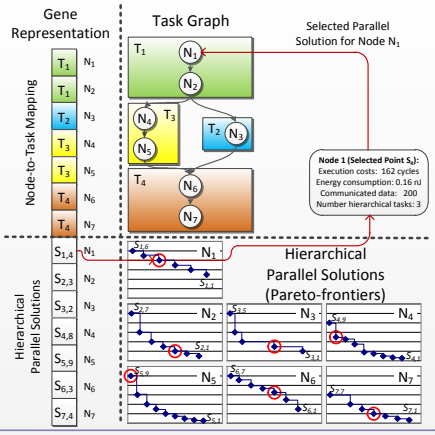


- Approach extracts more fine-grained pipeline-parallelism from loops and loop nests from sequential ANSI-C applications
- Many embedded applications have a streaming-oriented structure
- Loops can be divided into concurrently executed tasks
  - Horizontal splits: Move statements to tasks
  - Vertical splits: Split iterations of tasks
- Program Dependence Graph as IR
- Both splits extracted by ILP in one step
- Local optima are avoided
- Speedup of up to 1.9x, 2.9x, and 3.9x (on 2, 3, and 4 core architectures)

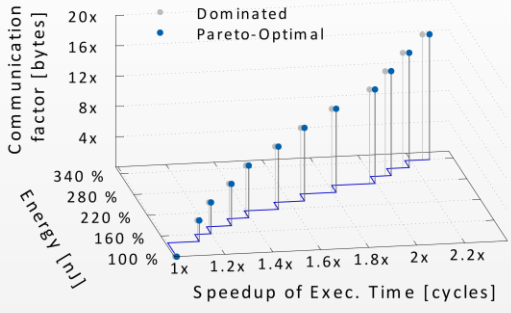


[2] Daniel Cordes, Andreas Heinig, Peter Marwedel, and Arindam Mallik. *Automatic Extraction of Pipeline Parallelism for Embedded Software Using Linear Programming*. In Proc. of ICPADS 2011.

Multi-Objective Aware Extraction of Task-Level Parallelism Using Genetic Algorithms



- Embedded systems must be optimized for multiple objectives
- Same kind of parallelism extraction as first approach
- But, this approach observes three objectives while parallelizing an application:
  - Execution time
  - Energy consumption
  - Communication overhead
- Reduce amount of extracted parallelism to e.g., save energy
- Also based on hierarchical task graph
- User gets a Pareto-front of optimal solutions
- Choose the best implementation for specific scenario



[3] Daniel Cordes and Peter Marwedel. *Multi-Objective Aware Extraction of Task-Level Parallelism Using Genetic Algorithms.* In Proc. of DATE 2012.

Future Work

- Current approaches are optimizing for homogenous systems
  → make them applicable for heterogeneous systems
- Adapt pipeline-parallelism to be multi-objective aware
- Test and apply different communication optimizations

Daniel Cordes | Peter Marwedel | Web: ls12-www.cs.tu-dortmund.de/~cordes | e-Mail: firstname.lastname@tu-dortmund.de

TU Dortmund University | Department of Computer Science XII | Otto-Hahn-Str. 16 | 44221 Dortmund | Germany