README

Multiobjective code compression and function inlining for hard real-time systems

DOI: 10.15480/336.4890

The dataset consists of raw data used to produce results for Muts's Ph.D. Thesis "Multiobjective compiler-based optimizations for hard real-time systems".

An embedded system is a computer system consisting of software and hardware that is dedicated to a specific task within a larger system. Modern embedded systems are small devices operating on batteries and having a limited memory space, so their important restrictions are the code size and energy consumption of embedded programs.

An embedded systems that must react before a given deadline is called a real-time system. If violation of timing constraints might lead to catastrophic consequences, the system is called a hard real-time system. The most important property of a hard real-time system is its worst-case execution time (WCET).

The thesis introduces a new compiler-based **compression technique** that compresses chunks of a program at compile time and decompresses them at runtime. The technique guarantees that the compiled program with the runtime decompression satisfies its timing requirements. For compression, FastLZ library was used.

Many embedded systems have to fulfil multiple contradicting design criteria. The thesis presents a new approach to solve **multiobjective problems at compile time** when considering worst-case execution time, code size, and energy consumption as objectives. Since evolutionary algorithms can find trade-offs between the objectives but only for small programs due to very time-consuming evaluations of worst-case execution time and energy consumption at compile time, the thesis introduces

1. a prediction model based on machine learning techniques to predict worst-case execution time and energy consumption instead of costly estimating them. For predictions, we used Scikit-learn: Machine Learning in Python;
2. a search space reduction technique to run evolutionary algorithms on smaller search spaces.

To demonstrate the advantages and disadvantages of evolutionary algorithms when solving multiobjective compiler-based optimization problems, the thesis demonstrates the results for a well-known compiler-based optimization called **function inlining**.

To generate the data, we used WCET-aware compiler framework WCC for the Infineon TriCore TC1797 micro-controller with 88K of data section and 39K of SPM (compression) and ARM Cortex-M0 (function inlining). We ran evaluations on a server with the following:

- CPU: Dual CPU Intel XEON Gold 6146;
- RAM: 1.48T;
- Number of CPU cores: 48;
- CPU frequency: 3.30GHz;
- Operation system: Ubuntu 18.04.4.LTS.

We computed worst-case execution time (WCET) and energy consumption by AbsInt's aiT Worst-Case Execution Time Analyzers and EnergyAnalyser 20.10i and code size by WCC. We used Gurobi Optimizer version 8.1.0 to solve ILPs.

We considered benchmarks from the following benchmark suites: JETBENCH, MRTC, MediaBench, PolyBench, UTDSP, linear-algebra, and misc with loop bounds annotated by TACLeBench project.

This work has received funding from Deutsche Forschungsgemeinschaft (DFG) under grant FA 1017/3-1 and FA 1017/3-2.

Authors

- Kateryna Muts k.muts@tuhh.de

  Hamburg University of Technology (TUHH)

  Institute of Embedded Systems

  http://www.tuhh.de/es
  ORCID iD: 0000-0002-2255-3410

Directories/Files

A list of the directories and files, followed by an associated comment:

- README: this file.
- data/compression (Chapter 5 of the thesis): compression results.
  - compression.csv: compression statistics.
  - compress_runtime.csv: compression runtime.
- data/funinlining: results of the multiobjective function inlining problem.
  - evolag (Chapter 6 of the thesis):
    - ait-runtime.csv: aiT runtime for each benchmark.
    - fixedParams: results after fixing the hyper-paramters of GDE3 and MBPOA.

        - GDE3Front.csv: Pareto fronts after soving the problem by [GDE3](GDE3).
        - MBPOAFront.csv: Pareto fronts after soving the problem by MBPOA.
    - origMBPOA/MBPOAFront.csv (Chapter 7-9 of the thesis): Pareto fronts after soving the problem by MBPOA; it was used to compare the results with the one from the directories below.
    - predictionsScikit (Chapter 7 of the thesis): the results when solving the problem by using MBPOA with predicited WCET and energy consumption.
        - WCETEnergyAdaBoost: using AdaBoost for predictions.
            - ait-runtime.csv: aiT runtime for each benchmark.
            - MBPOAFrontPredict.csv: Pareto fronts after soving the problem by MBPOA with predictions.
            - model statistics.csv: statistics of the prediction model.
            - predictRuntime.csv: prediction runtime.
        - WCETEnergyDecTree: using decision tree for predictions. The structure is similar to the directory WCETEnergyAdaBoost.
        - WCETEnergyLogReg: using logistic regression for predictions. The structure is similar to the directory WCETEnergyAdaBoost.
    - searchspacereduction (Chapter 8 of the thesis): the results when solving the problem by using MBPOA executed on a reduced search space.
        - ait-runtime.csv: aiT runtime for each benchmark.
        - keptIndices.csv:
        - MBPOAFrontReduceSP.csv: Pareto fronts after soving the problem by MBPOA with reduced search spaces.
        - scores.csv: statistics of the prediction model.
    - searchspacereduction+predictions (Chapter 9 of the thesis): the results when solving the problem by using MBPOA with predictions executed on reduced search spaces.
        - RPWCETEnergyAdaBoost: using AdaBoost for predictions.
            - ait-runtime.csv: aiT runtime for each benchmark.
            - MBPOAFrontPredict.csv: Pareto fronts after soving the problem by MBPOA with predictions executed on reduced search spaces.
            - model statistics.csv: statistics of the prediction model.
            - predictRuntime.csv: prediction runtime.
        - RPWCETEnergyDecTree: using decision tree for predictions. The structure is similar to the directory RPWCETEnergyAdaBoost.
        - RPWCETEnergyLogReg: using logistic regression for predictions. The structure is similar to the directory RPWCETEnergyAdaBoost.