ePRO-MP: energy PRofiler and Optimizer for Multi-Processor

Wonil Choi, Hyunhee Kim, Wook Song, Jiseok Song, and Jihong Kim
School of Computer Science and Engineering, Seoul National University, Seoul, Korea

Introduction

- Two main design goals for mobile embedded applications
  - High performance
  - Low energy consumption
- Requirements for development tools
  - Easy profiling of the performance and energy consumption of applications
  - Easy identifications of energy and performance bottlenecks in the application level

Motivations

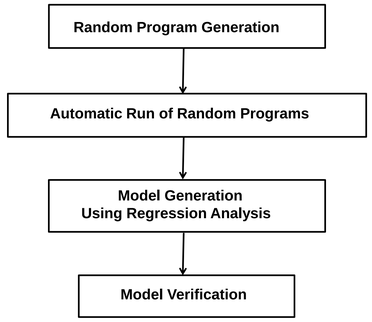
- Use of power measurement equipments for power profiling
  - Average embedded software developers are reluctant to use them
  - They are often too expensive to be widely used
- Absence of automatic optimization support
  - An efficient implementation often requires to explore a large design space
  - E.g., determining the optimal number of threads for co-running applications

Contributions

- Design of a system development tool, ePRO-MP
  - Profiles both performance and energy consumption of multi-threaded applications
  - Presents the analysis results in the program, thread, and function levels
- Energy profiling without requiring a power measurement equipment
  - Performance date and a regression-based energy model are used
- A limited support for automatic optimization functions
  - Two optimization examples are shown

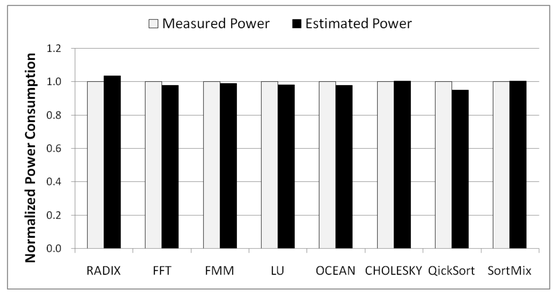
Model-Based Energy Profiler

- Power model development procedure



Procedure for Deriving Energy Model

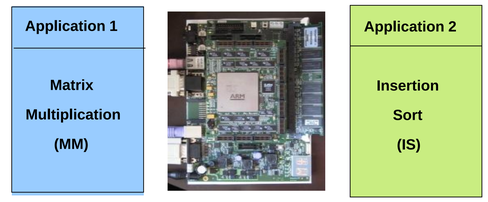
- Power model for ARM11 MPCore
  - Constraint: no per-core level power measurement
  - Power = A X (*Instr / time*) + B X (*DL1Access / time*) + C X (*L2Access / time*) D X (*DataDep / time*) + E X (*CohTrans / time*) + F_{const}
  - Average error 3% for SPLASH-2 benchmark programs



Comparison of Measured Power and Estimated Power

Profile-Based Automatic Optimizer

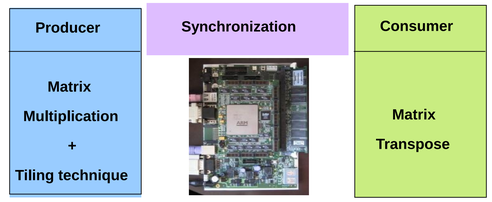
Optimization Example 1: Co-Running Applications Optimization



How many threads ? **4 cores** *How many threads ?*

- Results
  - Three threads for MM(A) and one thread for IS(B)
  - Performance improvement 6.1% and energy improvement 4.1%

Optimization Example 2: Producer-Consumer Application Optimization

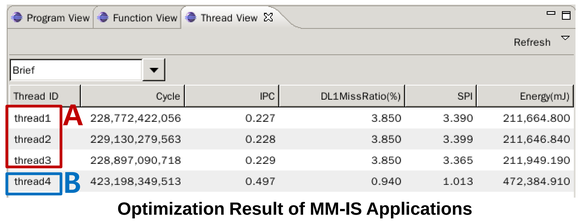


How many threads ? **4 cores** *How many threads ?*

How much tile size ? **32 KB $L1**

- Results
  - Four threads for MM, one thread for MT, and the tile size close to L1 cache size
  - Performance improvement 60.5% and energy improvement 43.3%

Optimized target software

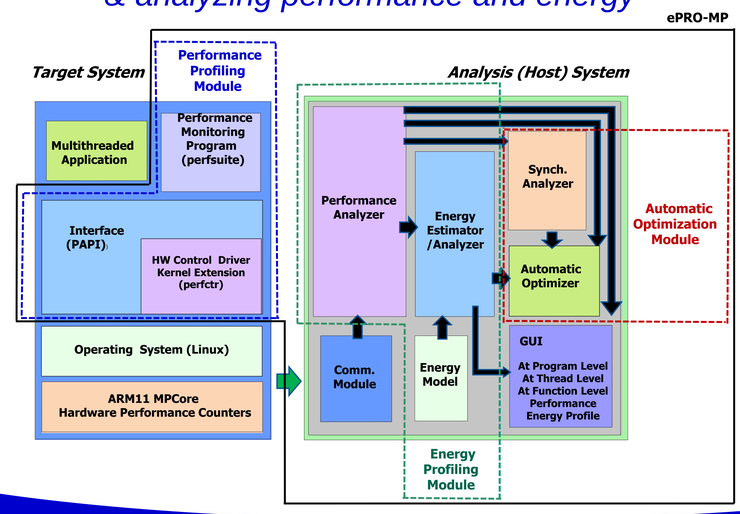


Optimization Result of MM-IS Applications

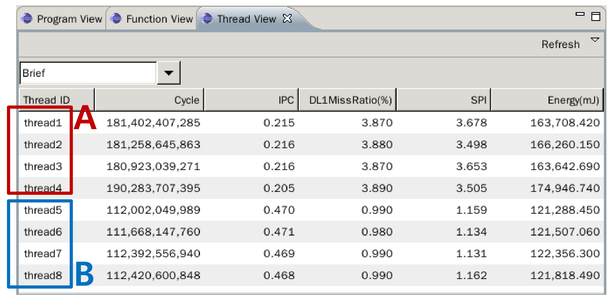
Optimization Result of MM-MT Application

Developing target software

Executing target software & analyzing performance and energy



Detecting bottlenecks and hot-spots



Profiling Result of (4, 4) Thread Allocation Shown in Thread Level