

Design Tools Session

ePRO-MP: energy PProfiler and Optimizer for MultiProcessors

Wonil Choi¹, Hyunhee Kim², Wook Song³, Jiseok Song⁴, and Jihong Kim⁵

¹*Seoul National University, Seoul, Korea, choi11@davinci.snu.ac.kr*

²*Seoul National University, Seoul, Korea, hh0726@davinci.snu.ac.kr*

³*Seoul National University, Seoul, Korea, answer03@davinci.snu.ac.kr*

⁴*Seoul National University, Seoul, Korea, danielsong@davinci.snu.ac.kr*

⁵*Seoul National University, Seoul, Korea, jihong@davinci.snu.ac.kr*

Abstract

For mobile multiprocessor applications, achieving both high performance and low energy consumption becomes a challenging task. In order to meet these design requirements, programmers should understand the performance and the energy consumption of their applications, thus making system development tools play an important role. We describe ePRO-MP which profiles and optimizes the performance as well as the energy consumption of multi-threaded mobile applications using the POSIX thread library running on top of Linux for ARM11 MPCore-based embedded systems.

The main feature of ePRO-MP is that it can accurately estimate the energy consumption of multi-threaded applications without a power measurement equipment, using a regression-based energy model. We proposed an energy model development procedure based on the linear regression analysis. Following the methodology, we derived an energy model for ARM11 MPCore and verified its accuracy using several benchmark programs. The results showed that the maximum error was about 5%.

The second key feature of ePRO-MP is an automatic optimization function. Automatic optimization support is becoming more important in multiprocessor-based embedded systems because an efficient implementation often requires exploring a large design space as the number of cores increases and the co-running threads share the limited system resources. The automatic optimization function of ePRO-MP helps programmers to improve the performance and energy consumption of two case examples without their intervention.

The experimental result showed that it improved the performance and energy consumption by 6.1% and 4.1%, respectively, over a baseline version for the co-running applications and also improved the performance and energy consumption by 60.5% and 43.3%, respectively, over a baseline version for the producer-consumer application.