Dynamic Phase-based Optimization of Embedded Systems

Tosiron Adegbija and Ann Gordon-Ross*

Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
tosironkbd@ufl.edu & ann@ece.ufl.edu
Also affiliated with the NSF Center for High-Performance Reconfigurable Computing (CHREC) at UF

Introduction and Motivation

- Embedded system optimization is challenging
  - Numerous tunable parameters (e.g., cache size, clock frequency, etc.)
  - Many combinations ➔ large design space ➔ long design space exploration time
  - Dynamic application execution requirements (i.e. *phases* of execution)
  - Need to specialize (tune) configurations to different application phases
- *Phase-based optimization* dynamically determines each phase's best configuration (i.e., tunable parameter combination)
  - Satisfies optimization goals (e.g., lowest energy, best performance)
- Determining each phase's best configuration is challenging
  - Potential for large runtime overhead (energy and/or performance)

Goal

Determine each phase's best configuration with minimal runtime overhead/design-time effort

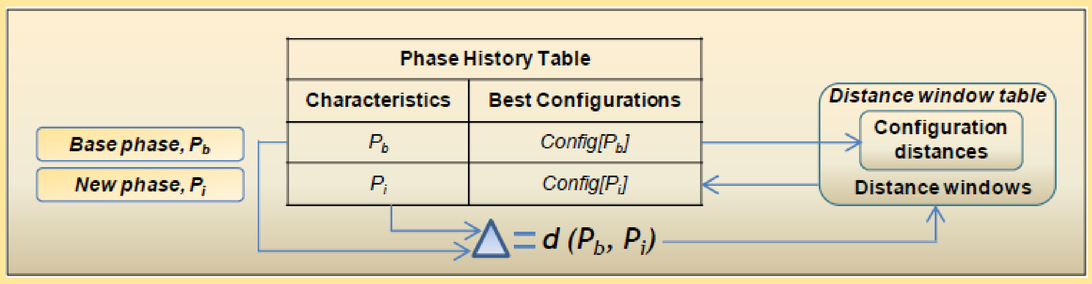
Major Contributions

- Propose ***Phase Distance Mapping (PDM)***
  - Low overhead, computationally simple, and dynamic method for determining a phase's best configuration
  - Significant reduction in optimization overhead compared to extensive design space exploration
  - Minimal designer effort
  - Applicability to various dynamic optimization scenarios
- Evaluate PDM in various optimization scenarios
  - ***Cache-tuning***
  - ***Thermal-aware phase-based optimization (TaPT)***

Phase Distance Mapping (PDM)

Approach

Tune only ***base phase*** P_b and predict new phase P_i's best configurations based on correlations of P_b's and P_i's characteristics (e.g., cache miss rates, instruction per cycle, etc.)



Phase distance mapping (PDM) overview

PDM overview

When new phase executed, compute *phase distance* $d(P_b, P_i)$

- Computing $d(P_b, P_i)$
  - **Normalization**, for single characteristics
  - **Euclidean distance**, for multiple characteristics
    E.g., to tune instruction cache, data cache, clock frequency:

$$d(P_b, P_i) = \sqrt{(iMR_{Pb} - iMR_{Pi})^2 + (dMR_{Pb} - dMR_{pi})^2 + (IPC_{Pb} - IPC_{Pi})^2}$$

iMR, *dMR*, and *IPC:* instruction cache miss rate, data cache miss rate, and instruction per cycle, respectively

Search *distance window table* for $d(P_b, P_i)$'s distance window

- ***Distance window***: phase distance range with minimum Win_L and maximum Win_U value

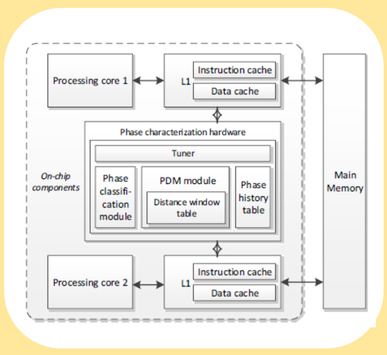
$$Win_L < d(P_b, P_i) < Win_U$$

- Contains configuration distance from P_b's best configuration

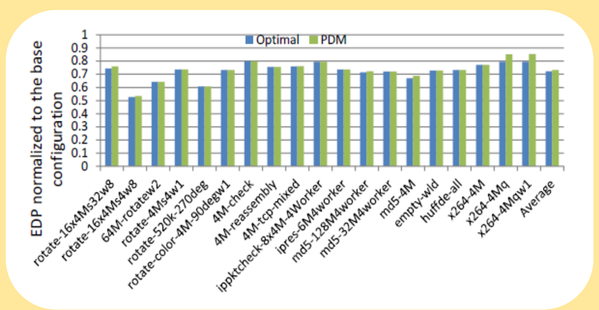
Update/store P_i's best configuration and execute P_i in that configuration

Results

Cache Tuning

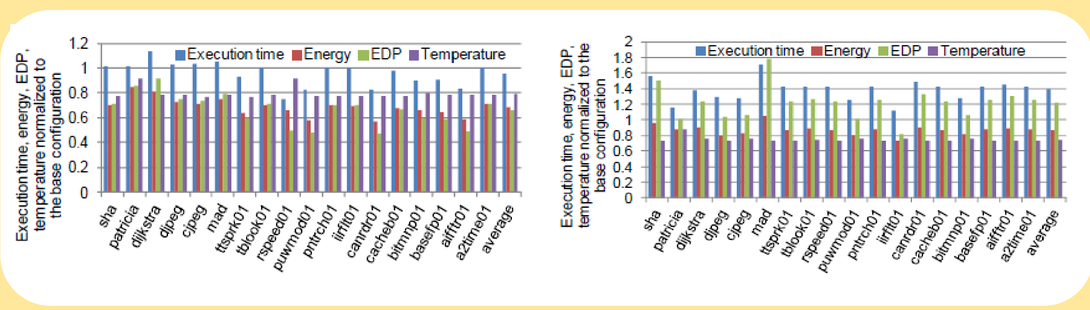


Phase-based cache tuning architecture for a sample dual-core system



EDP savings as compared to the base configuration

Thermal-aware Phase-based Tuning (TaPT)



Execution time, energy, EDP, and temperature normalized to the base configuration for priority settings energy [left] and temperature [right]

- PDM imposed minimal area/power overheads
  - 0.3% and 0.11% area and power overheads on MIPS M4K processor
- PDM cache configurations' achieved significant EDP savings as compared to base configuration
  - Average of 27%, with savings as high as 47%, within 1% of optimal
- TaPT determined Pareto optimal cache and clock frequency configurations based on user-specified settings
  - Priority settings: *execution time, energy,* and *temperature*
  - Execution time, energy, and temperature reductions up to 5%, 30%, and 25%, respectively

PDM Usage Scenarios

- Potential usage scenarios to motivate future research
  - Fast tuning of configurable hardware and multi-objective optimization scenarios
  - Scheduling in heterogeneous core systems
  - Speed up simulations in computer architecture research

Conclusions

- PDM enables designers to dynamically determine optimal/near-optimal configurations
  - Minimal runtime overhead and design-time effort
  - Suitable for various optimization scenarios and general purpose embedded systems