2015 年 7 月 13 日 情報システム基盤学合同輪講発表資料

Power-Performance Modeling on Asymmetric Multi-Cores

著者： *M. Pricopi, T. S. Muthukaruppan, & V. Venkataramani*
出典： *Int'l. Conference on Compilers, Architecture and Synthesis for Embedded Systems, p.1-10, 2013.*
発表者： 高性能コンピューティング学講座 本多・三輪研究室 1553007 澁谷俊憲

1 はじめに

非対称型マルチコアは、ヘテロジニアスマルチコアの一種であり、異なる種類のコアを搭載したプロセッサである。非対称型マルチコアは、一般に “処理性能に優れるが、高消費電力” の big コアと “電力効率に優れるが、比較的処理性能が劣る”small コアで構成されている。これらのコアに適切にタスクをスケジューリングすることで、性能を維持しつつ、高い電力効率を実現する。

2 先行研究

実行コアが不適切な場合、性能は低下し、消費電力が大きくなる。そのため、実行中に動的に実行コアを切り替える手法が考えられた。ただ、実行コアの切り替えにもオーバーヘッドがあるため、実際に実行コアを切り替えることなく、より適切なタスクの実行コアを推定する方法が考えられている。

Craeynest らは、一方のコアタイプ (big/small) 上でのアプリケーションの実行プロファイルから、他方のコア (small/big) の性能を推定するモデル (Performance Impact Estimation:PIE) を提案した [1]。しかし、このモデルは実在のプロセッサのみでは得られない情報を基にしており、実際にハードウェアに対して適用することが困難であった。また、処理時間の向上を目指した研究だったため、電力推定がなされていなかった。

本稿では、実際のプロセッサから得られる情報のみを用いたコア間性能推定モデルを提案する。

3 ARM big.LITTLE

本研究で対象とするハードウェアである ARM 社の big.LITTLE プロセッサには big コアとしてアウトオブオーダーコアの Cortex-A15 が 2 つ、small コアとしてインオーダーコアの Cortex-A7 が 3 つ搭載されている (図 1)。この 2 種類のコアはメモリ階層などの各種構造パラメータが異なっている。先行研究 [2] ではパイプライン構成などのコアの特性が同一である仮定のもとに行われた。

4 性能推定モデリング

性能をモデル化するにあたり、本研究は CPI(Cycles Per Instruction) に着目する。各コアに対し、必要クロックサイクル C を回帰的に推定した後、命令数 N で除することで実効的な CPI を推定できる。

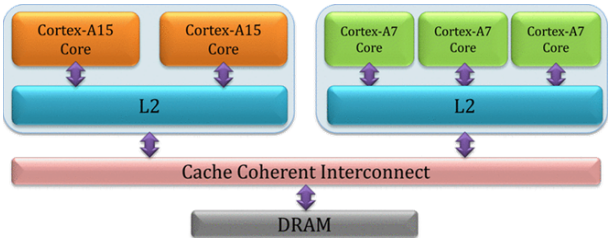


図 1: ARM big.LITTLE の構造

4.1 CPI スタック [2]

CPI を “パイプラインなどアーキテクチャ依存の要因”CPI_{steady} と “キャッシュミスなどの外的要因”CPI_{miss} の 2 種類が消費する部分サイクルに分けた CPI スタックとして考える (式 1)。

$$CPI = CPI_{steady} + CPI_{miss} \tag{1}$$

CPI_{miss} は、プログラムの入力に依存し、ミスイベントとその遅延で表現できる。CPI に特に影響を与えるミスイベントは、分岐予測ミス、L1、L2 キャッシュミスである。対して CPI_{steady} はコアタイプ、プログラムで固有の特性を持ち、プログラムへの入力に依存しない。よって、コンパイル時に解析可能であり、アプリケーションの基本ブロック B に対して、コア P の推定サイクル数 $cycle_B^P$、命令数 $instr_B$、推定ブロック出現頻度 $freq_B$ を抽出できる。これらの値を用いて、アプリケーション $\mathcal{A}$ に対する $CPI_{steady}^{\mathcal{A}}$ を式 2 に定義する。ブロックごとの値を総合することで、入力に依存しない平均的な値を算出できる。

$$CPI_{steady}^{P}\Big|_{\mathcal{A}} = \frac{\sum_B freq_B \cdot cycles_B^P}{\sum_B freq_B \cdot instr_B}\Big|_{\mathcal{A}} \tag{2}$$

4.2 big コアの CPI スタックモデル

big コアの CPI_{miss} は L1 キャッシュミス $miss_{L1I}$、分岐予測ミス $miss_{br}$、L2 データアクセスミス $dmiss_{L2}$ の影響を受ける。big コアのサイクル C^{big} の CPI スタックモデルは式 3 のように表現する。以後、各 β は回帰分析に用いられる係数である。

$$\begin{aligned} C^{big} = {} & \beta_0 \cdot N \cdot CPI_{steady} + miss_{L1I} \cdot c_{L2} \\ & + miss_{br} \cdot (c_{br} + c_{fe}) + dmiss_{L2} \cdot \frac{c_{mem}}{MLP} \end{aligned} \tag{3}$$

第 1 項は CPI_{steady} の寄与を表す。第 2、3、4 項について、c_{L2} は L2 キャッシュへアクセスした時間、c_{br} は分岐解

決時間中に消費した時間、c_{fe} はフロントエンド時に要した時間、c_{mem} はメモリアクセスによる消費時間に関連したサイクルである。L2 キャッシュミスは、アウトオブオーダーコアのメモリレベルの並列性 ($MLP = \beta_1 \cdot (dmiss_{L2}/N)^{\beta_2}$) によりレイテンシが軽減される。

4.3 small コアの CPI スタックモデル

small コアのサイクル数 C^{small} はインオーダーコアなので、式 3 からアウトオブオーダー固有の係数 (c_{br}、MLP) を削除したものとする (式 4)。

$$C^{small} = \beta_3 + \beta_4 \cdot N \cdot CPI_{steady} + miss_{L1I} \cdot c_{L2} + miss_{br} \cdot c_{f_e} + dmiss_{L2} \cdot c_{mem} \quad (4)$$

4.4 コア間ミス率推定

一方のタイプのコア P のミス率から他方のコア P' のミス率を推定する。

分岐予測ミスは分岐命令ミス率、CPI、命令あたり分岐数 N_{br}/N に依存している (式 5)。コア間 L1 命令キャッシュミスは初期化ミス $cold$ と衝突ミス $conflict$ によって発生する (式 6)。また、コア間 L1 データキャッシュミス ($miss_{L1D}$) はベンチマークを行ったところ、A7 と A15 で、ほぼ差がなかったため、単純な線形回帰とした (式 7)。

コア間 L2 キャッシュミスはデータアクセスミスのみがパフォーマンスカウンタに公開されているが、命令アクセスミスは CPI にほとんど寄与しない。これも L1 と同じく初期化ミス、衝突ミスで構成される (式 8)。ただし、L2 命令アクセスが多い場合はデータの追い出しが多くなるので、L2 の命令アクセス比で正規化している。

これらで得た値を式 3、4 に代入することで、他方のコアの CPI を推定 (コア間性能推定) することができる。

$$\overline{miss_{br}^{P'}} = \beta_5 + \beta_6 \cdot miss_{br} + \left(\frac{1}{CPI}\right)^{\beta_7} + \beta_8 \cdot \left(\frac{N_{br}}{N}\right)^{\beta_9} \Bigg|_P \quad (5)$$

$$\overline{miss_{L1I}^{P'}} = \beta_{10} + \beta_{11} \cdot cold \cdot \left(\frac{N}{N_{br}}\right)^{\beta_{12}} + \beta_{13} \cdot conflict \Bigg|_P \quad (6)$$

$$\overline{miss_{L1D}^{P'}} = \beta_{14} + \beta_{15} \cdot miss_{L1D} \Big|_P \quad (7)$$

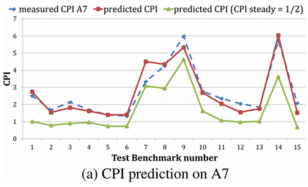
$$\overline{dmiss_{L2}^{P'}} = \beta_{17} \cdot conflict \cdot \frac{miss_{L1I}}{miss_{L1I} + miss_{L1D}} + \beta_{18} \cdot cold \Bigg|_P \quad (8)$$

5 電力モデリング

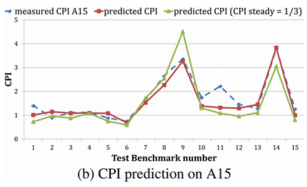
アーキテクチャ内の電力は構成要素に対して加算的であるため、電力モデルは線形回帰で分析可能である。

big コアの消費電力はパイプライン動作とメモリ階層へのアクセスにより決定される。式 9 中の前 3 項が命令 (N_{int}、N_{fp}、CPI)、後 3 項がメモリ階層へのアクセス ($access_{L1D}$、$access_{L2}$、$dmiss_{L2}$) に関する項である。この式に前節のコア間ミス推定から得た値を代入することで他方のコアの電力推定 (コア間電力推定) が可能となる。また、small コアはベンチマーク間で平均消費電力の差はほぼなく、モデル化は不要と判断した。

$$Power = \beta_{19} + \beta_{20} \cdot \frac{N_{int}}{N} + \beta_{21} \cdot \frac{N_{fp}}{N} + \beta_{22} \cdot \frac{1}{CPI} + \beta_{23} \cdot \frac{access_{L1D}}{N} + \beta_{24} \cdot \frac{access_{L2}}{N} + \beta_{25} \cdot \frac{dmiss_{L2}}{N} \quad (9)$$



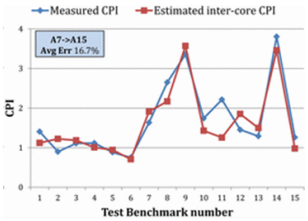
(a) CPI prediction on A7

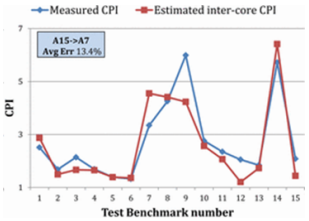


(b) CPI prediction on A15

[1]A7 [2]A15

図 2: CPI 推定値と測定値、先行研究の推定 [2] との比較





[1]A7 → A15 [2]A15 → A7

図 3: 各種ワークロードでのコア間性能推定

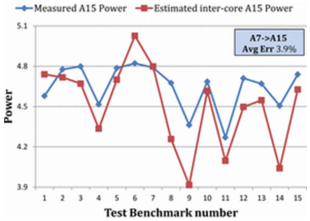


図 4: 各種ワークロードでのコア間電力推定 (A7 → A15)

6 評価

15 種類のベンチマークによるモデルの誤差評価を行った (図 2)。平均フィッティング誤差は A7 で 8.2%、A15 で 10.1%だった。先行研究 [2] との比較では A7 で 33.3%、A15 で 8.1%誤差を軽減できた。small コアのモデルは項が少ないため、精度が高かったと考えられる。コア間性能推定は A7 から A15 では、平均 13.4%、最大 43.2%、A15 から A7 では平均 16.7%、最大 41.3%だった (図 3)。

A7 から推定した A15 電力推定の平均誤差は 3.9%であった (図 4)。性能推定モデルのメモリアクセス部分の改善により高精度を導けたものと考えられる。

7 まとめ

非対称マルチコアアーキテクチャの性能、消費電力を推定するモデルを開発した。これは単一命令セットのヘテロジニアスマルチコア上で、一方のコアの実行プロファイルから、もう一方のコアのアプリケーションの電力性能の挙動を推測するものである。

これは従来の研究と異なり、実際のハードウェア (ARM big.LITTLE) の公開情報のみを用いた予測モデルである。

参考文献

[1] Kenzo Van Craeynest,et al. Scheduling heterogeneous multi-cores through performance impact estimation (PIE). ISCA, pages 213-224, 2012.

[2] Stijn Eyerman, Kenneth Hoste, and Lieven Eeckhout. Mechanistic-empirical processor performance modeling for constructing CPI stacks on real hardware. ISPASS, pages 216-226, 2011.