1A-02

組込み機器向け性能予測手法の一検討

堀井　圭祐†　　山本　整†　　桐村　昌行†
三菱電機株式会社　情報技術総合研究所†

1. はじめに

組込み機器向けのアプリケーションは PC でソースコードの実装とコンパイルを行い，生成されたバイナリを組込み機器に書き込み，動作検証するという開発プロセスが一般的である．このような開発プロセスにおいては性能要求を満たすことができない場合，組込み機器でボトルネックを解析する必要がある．しかしながら，組込み機器ではリソース制約などにより使用可能なツールが限られているため，性能解析に時間を要してしまう．

一方で，S/W と H/W の並行開発などを目的に PC にアプリケーションの動作環境を構築し，動作検証するという開発プロセスも普及している．PC では解析ツールが充実しており，効率的にボトルネックを解析できることから，性能評価も PC で行うことで開発期間の短縮が期待できる．この時，組込み機器でのアプリケーション実行性能（以下，実行性能）を予測することが課題である．

本稿では，PC でのアプリケーション実行結果から組込み機器での実行性能を予測する方式について検討した結果を述べる．

2. 関連技術

組込み機器での実行性能予測技術としては，命令セットシミュレータ（Instruction Set Simulator: ISS）を活用した方式が検討されている [1]．しかし，このような方式においてはターゲットの CPU アーキテクチャに対応した ISS が必要である．そのため，ISS が開発されていない CPU アーキテクチャについては実行性能を予測できないという問題がある．

そこで，多様な CPU アーキテクチャに対応できるよう ISS を利用せずに実行性能を予測する方式を検討する．

An examination of performance prediction for embedded systems
Keisuke Horii, Hitoshi Yamamoto, Masayuki Kirimura
† Information Technology R&D Center, Mitsubishi Electric Corporation

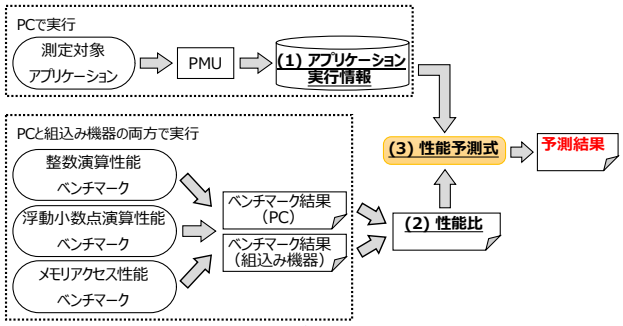

図 1　性能予測手順

3. 方式検討

PC に搭載されている CPU には CPU 内部で発生したパフォーマンスイベントを監視・記録するための Performance Monitoring Unit（以下，PMU）が存在すると想定する．PMU を利用すれば，対象アプリケーションの実行命令種別や実行回数などを取得することが可能である．そこで，PMU により取得したアプリケーション実行情報に基づく性能予測方式を提案する．

提案方式における性能予測手順を図 1 に示す．図 1 に示すようにアプリケーション実行情報と，ベンチマーク結果から算出した性能比を利用して組込み機器での実行性能を予測する．図 1 における(1)～(3)の詳細を次に示す．

(1) アプリケーション実行情報
PMU で取得する情報を表 1 に示す．

表 1　PMU 取得情報

#	取得情報
1	実行サイクル数（n_{Cycle}）
2	実行命令数（N）
3	単精度浮動小数点演算命令数（n_{Sfp}）
4	倍精度浮動小数点演算命令数（n_{Dfp}）
5	LOAD 命令数（n_{Load}）
6	STORE 命令数（n_{Store}）

(2) 性能比
性能比は “整数演算（R_{Int}）”，“単精度浮動小数点演算（R_{Sfp}）”，“倍精度浮動小数点演算（R_{Dfp}）”，“LOAD 処理（R_{Load}）”，“STORE 処理（R_{Store}）” の 5 種類算出する．

(3) 性能予測式

組込み機器での予測実行時間を T_{EMB}，PC での実行時間を T_{PC}，クロックサイクルを t，非取得命令数を n_{Other} と定義した場合の性能予測式を次に示す．

$$T_{\mathrm{EMB}} = T_{\mathrm{PC}}\left\{\left(\frac{n_{\mathrm{Sfp}}}{N}R_{\mathrm{Sfp}}\right)+\left(\frac{n_{\mathrm{Dfp}}}{N}R_{\mathrm{Dfp}}\right)+\left(\frac{n_{\mathrm{Load}}}{N}R_{\mathrm{Load}}\right)+\left(\frac{n_{\mathrm{Store}}}{N}R_{\mathrm{Store}}\right)+\left(\frac{n_{\mathrm{Other}}}{N}R_{\mathrm{Int}}\right)\right\}, \quad (1.1)$$

$$n_{\mathrm{Other}} = N-\left(n_{\mathrm{Sfp}}+n_{\mathrm{Dfp}}+n_{\mathrm{Load}}+n_{\mathrm{Store}}\right), \quad (1.2)$$

$$T_{\mathrm{PC}} = tn_{\mathrm{Cycle}}. \quad (1.3)$$

4. 評価

4.1. 評価環境

評価環境を表 2 に示す．測定対象のアプリケーションには組込み機器固有の I/O へアクセスしないものを選択し，OS 環境も一致させることで，PC で動作させるための改変は不要とする．また，PMU から各種情報を取得するために，プロファイリングツールの perf を利用する．

表 2 評価環境

H/W	CPU PC: Intel Core i7-9750H 2.6GHz×12 コア 組込み機器: ARM Cortex-A57 1.5GHz×4 コア
S/W	OS Linux 4.15.0 (PC)/4.14.75 (組込み機器), RootFS: Ubuntu 16.04 LTS (PC/組込み機器) ベンチマークツール 整数演算性能：CoreMark 浮動小数点演算性能：Whetstone メモリアクセス性能：sysbench PMU アクセスツール perf 測定対象アプリケーション 自動運転向け自車経路抽出プログラム

ベンチマーク結果から算出した組込み機器に対する PC の性能比を表 3 に示す．メモリアクセスの性能比はアクセスパターン（シーケンシャル/ランダム）に依存し，特に LOAD 命令についてはキャッシュアクセス（ヒット/ミス）の影響も大きいため，複数のパラメータを用いて実行性能を予測する．

表 3 性能比（PC 測定値/組込み機器測定値）

整数演算性能		2.23	
単精度浮動小数点演算性能		2.69	
倍精度浮動小数点演算性能		2.27	
メモリアクセス性能		ヒット時	ミス時
シーケンシャル	LOAD 処理	2.55	6.88
	STORE 処理	2.32	
ランダム	LOAD 処理	3.76	10.58
	STORE 処理	6.61	

4.2. 結果と考察

評価結果を図 2 に示す．図 2 における予測値は，表 3 の各性能比を性能予測式に適用して算出し，メモリアクセスの性能比のみ変更した 4 通りの結果を示す．また，予測精度を評価するために組込み機器での実測値もあわせて記載する．

図 2 の評価結果より，メモリアクセスの性能比として seq:miss と rnd:hit を適用した場合の予測誤差は 10%未満であり，高精度に予測できている．一方，seq:hit と rnd:miss を適用した場合は予測誤差が 40%以上となり，精度が低くなる．これは，測定対象のアプリケーションにおける LOAD/STORE 命令の実行比率は約 54%であり，予測結果に与える影響が大きいためである．つまりメモリアクセスの特徴を把握し，適切なメモリアクセス性能比を選択することで高精度な性能予測が可能になると考えられる．

PC のキャッシュミス率は PMU により算出可能であるが，性能予測のためにはキャッシュアーキテクチャの違いを考慮する必要がある．アクセスパターンについては，LOAD/STORE 命令実行をトレースし，アクセス先のアドレスを解析する方法などが考えられる．

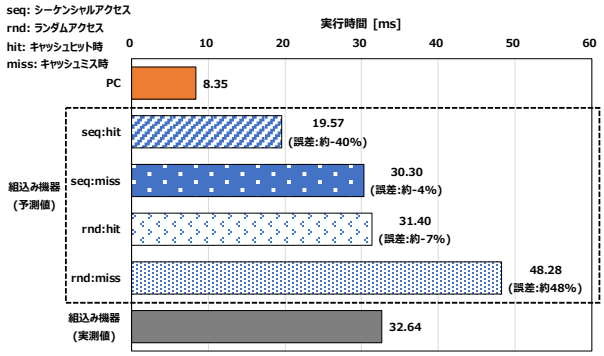

図 2 評価結果

5. おわりに

今回，PMU で取得可能なアプリケーション実行情報に着目した性能予測方式を検討した．提案方式を評価したところ，最高で誤差 4%未満の予測が可能であることを確認した．

今後は，より高精度な予測を実現するためにメモリアクセスの特徴や，分岐予測など他の命令に関する性能比も考慮した方式を検討する予定である．

参考文献

[1] 小山他，「性能評価のためのマルチコアシミュレーション方式の検討」，電子情報通信学会 2016.3