

# Predicting Performance Impact of DVFS for Realistic Memory Systems

著者: Rustam Miftakhutdinov, Eiman Ebrahimi, Yale N. Patt

出典: *Proceedings of the International Symposium on Microarchitecture (MICRO '12)*, pp.155-165, 2012.

発表者: 高性能コンピューティング講座 本多・近藤研究室 1353030 松村 正隆

## 1 はじめに

プロセッサの省電力化技術として、計算機の負荷状況に応じてクロック周波数や電圧を動的に変化させる DVFS (Dynamic Voltage and Frequency Scaling) がある。DVFS は DRAM やプロセッサにかかる負荷状況を正確に把握することで、効果的にプロセッサを省電力化することができる。しかし従来の手法では実際の DRAM が持つレイテンシの特性などを考慮に入れておらず、結果として削減可能な消費電力に対して低い割合でしか省電力化を達成できていない。そこで本研究では DRAM のレイテンシとプリフェッチの影響を考慮した新たな性能予測手法である CRIT+BW を考案し、DVFS 向けの正しい性能予測を図るものである。

## 2 従来の性能予測手法

従来の性能把握手法として、Leading loads[1] と Stall time[2] が提案されている。それらは次の前提を基にしている。1) 全ての DRAM は同一のレイテンシである、2) DRAM へのアクセスが行われた際、プロセッサは処理を継続できるが、DRAM からの処理結果が帰る前にやがてストールする。これらの理論から得られる DVFS の性能モデルを図 1 に示す。計算機の消費電力は DRAM やプロセッサといった動的に変化するものと、ファンのように常に一定なものがある。DVFS を適用する上で、それぞれ性質の異なるこれらの消費電力を考慮し、総合的に最も低くなるように周波数、電圧を設定しなくてはならない。例えば動的な消費電力はクロックと電圧を下げることで削減できるが、その分処理に時間がかかり常に一定の電力がかかる機器のためにかえって総消費電力は上がってしまう可能性がある。そのため図 1 で示されている DRAM

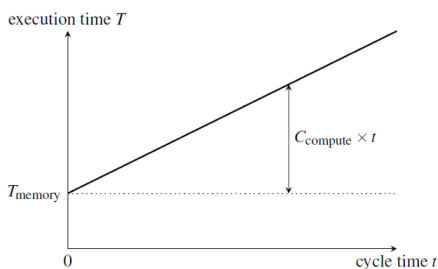


図 1: リニア DVFS 性能モデル

とプロセッサの動作時間を正確に求める必要がある。実行時間  $T$  は DRAM 駆動時間  $T_{\text{memory}}$  とプロセッサの実行時間の和で求めることができる。プロセッサが駆動したクロック数を  $C_{\text{compute}}$ 、クロック周期を  $t$  とすると、

$$T(t) = T_{\text{memory}} + C_{\text{compute}} \times t \quad (1)$$

となる。そこで式 1 から  $C_{\text{compute}}$  を求めるために  $T_{\text{memory}}$  をハードウェア上で測定する必要があるが、従来手法ではそれぞれ異なる方法を用いている。Leading loads はプロセッサが動作してストールするまでの区切りを 1 作業単位として、DRAM の Load 命令が出された作業単位の個数とレイテンシの積で  $T_{\text{memory}}$  を求めており、Stall time はプロセッサのストール時間を DRAM 稼働時間と近似して求めている。

## 3 従来手法の問題点

従来の性能モデルには、2 つの問題点がある。1 つ目に DRAM のレイテンシはすべて同一ではない点がある。DRAM からのデータ読み出しの状況として、異なる DRAM スロットからの読み出しと同一スロットからの読み出しの場合がある。異なるスロットからの読み出しの場合、データはそれぞれのスロットから並列にプロセッサに伝送される。それに対して同一スロットからの場合は逐次的に伝送されるため、速度に差が出ることになる。更に DRAM は行と列のアレイ構造によってデータを保持しているが、それを読み出す際は行単位でアクセスし、それをバッファに保存するため、連続するアクセスが異なる行からの読み出しか同行からの読み出しでもアクセス遅延には差が出る。

2 つ目にプロセッサのプリフェッチ機能をオンにした場合について考える。このときクロック周波数を上げ続けてプロセッサの実行速度を速めても、DRAM の帯域がボトルネックとなるため実行速度とクロック周波数は線形にならない。しかし、これは図 1 で示した性能モデルと矛盾するため、DRAM 帯域の限界についても考慮する必要がある。

## 4 DRAM 性能の正確なモデル化手法

### 4.1 CRIT

問題点の 1 つ目に挙げた DRAM のレイテンシが不均一であることを考慮すると、DRAM の動作時間は Leading

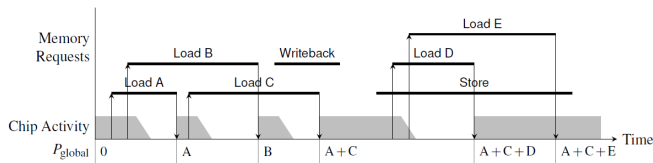


図 2: 非同一レイテンシの DRAM とプロセッサの様子

loads の手法を用いて求めることはできない。そこで、“クリティカルパス”の概念を導入し、それを DRAM 動作時間とする。クリティカルパスはプロセッサのクロック周期を 0 に近似した際の実行時間、すなわち図 1 の y 切片を指す。更にプロセッサの作業単位が異なる 2 つのロード (A,B) があるとき、A のデータが B のロード命令が出される前に返った場合、その 2 つのロードは“シリアル化”されていると定義する。シリアル化されているロードはそれが一塊のロード時間とみなし、最終的にクリティカルパスを求める際に用いられる。これを説明するため図 2 を示す。図 2 の例で説明すると、図中で Load A、Load C がシリアル化されており、同じタイミングで呼ばれた Load B と比較して A+C のロード時間が長いいため、それがクリティカルパスのカウンタ ( $P_{global}$ ) に入力される。更に次の Load D、Load E も同じ論理で比較すると、クリティカルパスは A+C+E となり、メモリ動作時間が正しく求められていることがわかる。

## 4.2 BW

次に問題点の 2 つ目を解決するため、BW と呼ぶ解決手法を提案する。新たな DVFS 性能モデルを図 3 に示す。図 3 中、 $T_{demand}$  はプリフェッチに関係無い処理 (Store や Writeback など) の累計稼働時間、 $T_{memory}^{min}$  は DRAM が全ての命令の処理にかかる最小の時間、 $C_{compute}$  と  $t$  はプロセッサが駆動したクロック数とクロック周期である。これから  $C_{compute}$  を求めるために、以下の式を解く。

$$T(t) = T_{demand} + C_{compute} \times t + T_{prefetch\ stall}(t) \quad (2)$$

$T_{demand}$  は 4.1 節で説明した CRIT を用いることで求められる。 $T_{prefetch\ stall}(t)$  は DRAM のプリフェッチが追い付かず、プロセッサがストールした時間である。そのため新たなハードウェアカウンタを用意し、その時間を測定できる機能を用意した。これより  $C_{compute}$  を求めることができる。次に DRAM の動作時間を求める。これは  $T_{memory}^{min}$

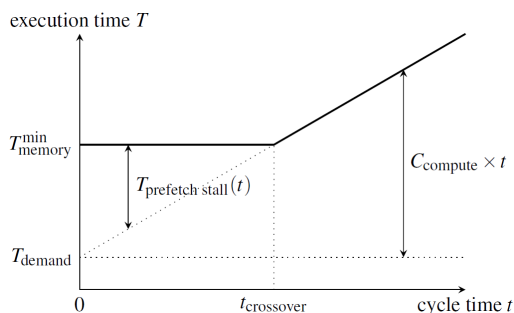


図 3: プリフェッチと DRAM 帯域を考慮した新モデル

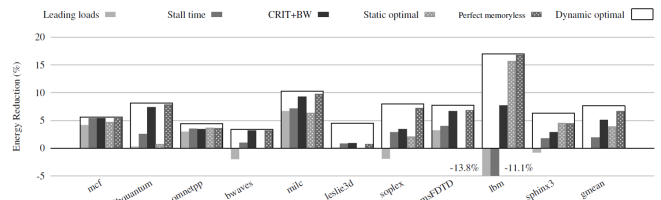


図 4: ベンチマークによる電力削減の解析結果

と等価であるため、以下の式で示すことができる。

$$T_{memory}^{min} = T(t) + T_{memory\ slack}(t) \quad (3)$$

式 3 の DRAM の非稼働時間である  $T_{memory\ slack}(t)$  もハードウェアカウンタを実装することで測定できるため、これより DRAM の稼働時間を求めることができる。

## 5 結果

解決手法で示した CRIT+BW を用いてシミュレータ上で計測を行い、消費電力の削減率を測定した結果を図 4 に示す。DVFS の設定として  $10^4$  個のインストラクション毎にプロセッサのクロック周波数を 1.5-4.5GHz から 100MHz 刻みで電圧・周波数を変更可能とした。ベンチマークは 16 種を実行し、それぞれにおいて従来手法と提案手法、最大削減可能量を測定した。最大削減可能量は  $10^4$  個のインストラクション毎に最も低消費電力となるクロック周波数で実行した際の削減量とした。図 4 より、削減可能量の平均値が 7.6%、Leading loads が 0.1% 以下、Stall time が 1.8% に対し、提案手法は平均値で 5% の電力削減を実現した。

## 6 まとめ

従来の性能予測手法はそれぞれのロードのレイテンシが異なることや、プリフェッチや DRAM の帯域の影響が考慮されていないことを示した。それに対する解決手法として CRIT+BW を提案し、シミュレータ上で評価を行った。評価結果では、従来手法と比較して電力が削減できることを示した。

## 参考文献

- [1] B.Rountree, D.K.Lowenthal, M.Schulz, B.R.de Supinski, “Practical Performance Prediction Under Dynamic Voltage Frequency Scaling”, IGCC '11 Proceedings of the 2011 International Green Computing Conference and Workshops (2011), pp.1-8.
- [2] Georgios Keramidas, Vasileios Spiliopoulos, Stefanos Kaxiras, “Interval-Based Models for Run-Time DVFS Orchestration in SuperScalar Processors”, ACM international conference on Computing frontiers (2010), pp.287-296.