

# MemScale: Active Low-Power Modes for Main Memory

著者： Qingyuan Deng, David Meisner, Luiz Ramos, Thomas F. Wenisch, Ricardo Bianchini,  
 出典： ASPLOS '11, Pages 225-238  
 発表者： 1553002 石原雅也

## 1 はじめに

ここ数十年の間、メインメモリとして使用されている DRAM(Dynamic Random Access Memory) は、世代が新しくなるにつれ電力効率は改善されつつある。その一方で、実装密度の向上やバンド幅の向上によって、メモリの消費電力は増加傾向にある。これは、消費電力の半分近くをメインメモリが占めているサーバにとって大きな負担となっている。そのため、メインメモリに関する省電力化が必要になっている。

## 2 研究背景

メインメモリの省電力化に関する手法のひとつに、スリープモードを活用したものがある。これは、メモリアクセス間隔を長くすることでアイドルタイムを長くし、その時間にメモリをデータ保持のために必要最小限のリフレッシュのみを行うモードで動作させている。しかし、モードの移行による電力の消費から、十分なアイドルタイムが確保できない場合は効果が低くなってしまふ。また、モード移行時に発生する時間のオーバーヘッドのために、プロセッサの実行時間が増加してしまう。

そこで本論文では、CPU の省電力化に用いられている DVFS(Dynamic Voltage and Frequency Scaling) をメモリに適用している。DVFS は、動作周波数と電源電圧を下げることで消費電力を下げるため、長いアイドルタイムを必要とせず、またオーバーヘッドも小さい。一方で、動作周波数を下げると処理速度も下がりプロセッサの実行時間が増加してしまうという課題が残る。この問題を解決するために、プロセッサの実行時間を予測することで、プロセッサの実行時間の増加を一定以内に抑えている。

## 3 MemScale

本論文では、メモリに DVFS 制御を活用し、パフォーマンスとエネルギーの予測モデルをたてることで、性能低下を一定以内に抑えることのできる DVFS 制御を提案している。

### 3.1 MemScale の実行

MemScale の実行例を図 1 に示す。ここで、 $T_{Target}$  : 目標実行時間、 $T_{Actual}$  : 実際の実行時間、Max Frequency :

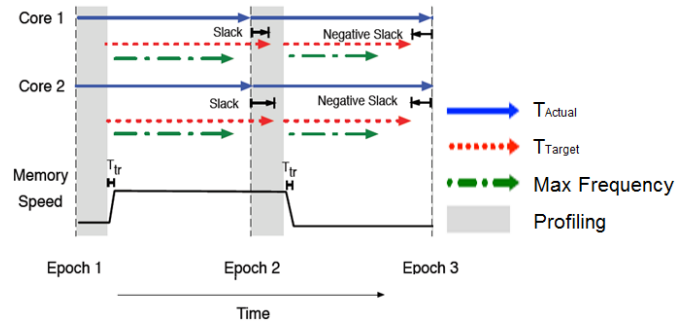


図 1 MemScale の実行例

メモリが最大周波数の時の予測時間、Profiling : 目標時間と予測時間の算出フェーズとする。

DVFS を適用する間隔を Epoch で分けて、それぞれの Epoch 内で計測、予測、適用を行っている。まず、プロファイリングフェーズにおいて、メモリが最大周波数で動作した場合のコアの予測実行時間を推定し、その後、DVFS による性能低下を見込んだコアの目標実行時間を設定する。図 1 では、Epoch1 の終了時、実際の実行時間が目標よりも早く終わっているため、メモリの速度をもっと低下させてもよいことがわかる。そのため次の Epoch2 は、メモリの速度を遅くして実行される。ここでの余裕を Slack として表し、各 epoch の終了時に測定する。

### 3.2 Slack

MemScale では、Slack を用いた性能の調整を行う。最大周波数と性能制約から目標実行時間を定め、実際の実行時間からみてどれだけの余裕があるのかを Slack として定義している。ここで、 $\gamma$  : 許容性能低下率 ( $\gamma \geq 0$ ) とする。

$$\begin{aligned} \text{Slack} &= T_{\text{Target}} - T_{\text{Actual}} \\ &= T_{\text{MaxFreq}} \cdot (1 + \gamma) - T_{\text{Actual}} \end{aligned}$$

### 3.3 エネルギーの予測モデル

システムのエネルギー比 (SER) を次式のモデルで定義し、用意した 10 個の周波数 (800, 733, 667, 600, 533,

467, 400, 333, 267, 200MHz)の中からSERが最小の値をとるときの周波数を選択する。ここで、 $T_{f_{Mem}}$  : 周波数  $f_{mem}$  のエポックでのパフォーマンス推定値、 $P_{f_{Mem}}$  : 周波数  $f_{mem}$  における電力推定値、 $T_{Base}, P_{Base}$  : 標準周波数に基づくパフォーマンス、電力とする。 $T_{f_{mem}}$  は次項モデルから推測し、 $P_{f_{mem}}$  は先行研究から求める [1]。

$$SER(f_{mem}) = \frac{T_{f_{Mem}} \cdot P_{f_{Mem}}}{T_{Base} \cdot P_{Base}}$$

### 3.4 パフォーマンスの予測モデル

MemScale ではパフォーマンスの予測モデルを、プログラムの平均 CPI を用いて次のように定義している。ここで、 $E[TPI_{CPU}]$  を CPU で費やす平均時間、 $E[TPI_{Mem}]$  をメインメモリへの LLC (Last Level Cache) ミス平均時間、 $\alpha$  を LLC ミスした割合、 $F_{cpu}$  をプロセッサの動作周波数としている。

$$E[CPI] = (E[TPI_{CPU}] + \alpha \cdot E[TPI_{Mem}]) \cdot F_{CPU}$$

メモリの周波数の変化による影響は、メインメモリで費やされる LLC ミスによる時間  $E[TPI_{Mem}]$  のみ受けるものとしてパフォーマンスの予測を行った。

## 4 評価

本論文で提案した MemScale は、ハードウェアで利用できない機能を用いているため、シミュレーションによってエネルギー削減率 (図 2) とパフォーマンス低下率 (図 3) の評価を行った。今回は、SPEC 2000 and SPEC 2006 suites [2] のアプリケーションを組み合わせることで、計算集約型 (ILP)、バランスのとれたワークロード (MID)、メモリ集約型 (MEM) のワークロードをそれぞれ用意した。また、データセンターにおける消費電力削減割合を算出するために、メモリシステムの平均電力消費率を全体の 40% と仮定 [3] し、残りの 60% をメモリ以外の固定平均電力推定値として計算を行った。

### 4.1 エネルギー削減率

計算集約型 (ILP) ではメモリシステムに関しては 70% 近くを、システム全体でも 30 パーセントほどのエネルギーが削減できている。これは、メモリアクセスがあまり発生しないため、メモリシステムの電圧と周波数を低い値で維持できていたからだと考えられる。対して、メモリ集約型 (MEM) では、計算集約型 (ILP) と比べ 1/3 程度の削減率にとどまった。これは、メモリアクセスが頻繁に発生することによってメモリシステムの電圧と周波数を低い値に維持できなかったためだと考えられる。

### 4.2 パフォーマンス低下率

計算集約型 (ILP) では性能低下はほとんどみられず、メモリ集約型 (MEM) では、大きなパフォーマンスの低下が

起こっていることがわかる。しかし、すべてのワークロードにおいて性能制約範囲に収まっていることが確認できる。

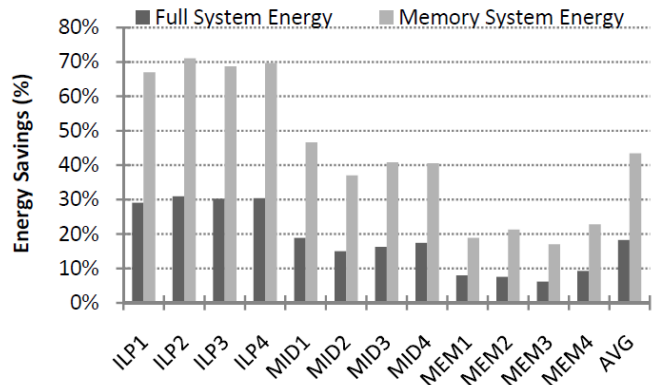


図 2 エネルギー削減率

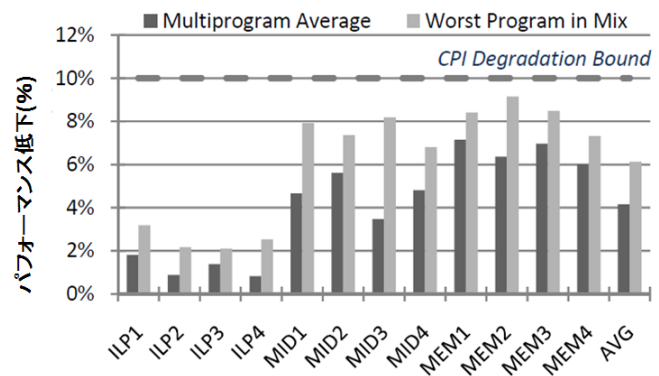


図 3 パフォーマンス低下率

## 5 まとめ

本稿では性能制約を設定したメモリへの DVFS 制御法 MemScale の提案を行った。そのために、パフォーマンスとエネルギーのモデル化を行い、収集したデータからの予測を行うことで動的な DVFS 制御を可能とした。結果、全てのワークロードで設定した性能制約を満たし、かつ平均 4.2% の性能低下で、18.3% の省電力化が可能となった。

## 参考文献

- [1] Micron. Calculating Memory System Power for DDR3, July 2007.
- [2] H. Zheng, J. Lin, Z. Zhang, and Z. Zhu. Decoupled DIMM : Building High-Bandwidth Memory System Using Low-Speed DRAM Devices. ISCA '09: International Symposium on Computer Architecture, 2009.
- [3] L. A. Barroso and U. Hölzl. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. Synthesis Lectures on Computer Architecture, Jan. 2009.