

Analysis of Dynamic Power Management on Multi-Core Processors

著者： W.Lloyd Bircher and Lizy K. John

出典： *Proceedings of the 22nd annual international conference on Supercomputing (ICS 2008)*, pp.327-338

発表者： 高性能コンピューティング講座 本多・近藤研究室 1053002 井上功一

1 概要

マイクロプロセッサの性能向上は、半導体の微細化やクロック周波数の向上といった手法からマルチコアを用いた並列処理による手法が主流となっている。マルチコアプロセッサにおいても省電力化は重要であるが、キャッシュコヒーレンスなどの問題からシングルコアに比べてより複雑なものとなっている。

本論文ではプロセッサの動作周波数に注目し、シミュレーションではなく、マルチコアを搭載する実機上で性能と消費電力を調査する。約6%の性能低下を伴うが、47%の消費電力を削減できることがわかった。

2 プロセッサの消費電力 [1]

プロセッサの消費電力は大きく分けて2つの要因がある。第1が動的な消費電力 (Dynamic Power) であり、第2が静的な消費電力 (Static Power) である。

動的な消費電力は実際にプロセッサが計算を行う際に生じる消費電力であり、トランジスタのスイッチング時に電荷の充放電に使われる電力である。動的な消費電力 P_d は以下の式 (1) のように表すことができる。

$$P_d = N_a C V^2 f \quad (1)$$

静的な消費電力はトランジスタが動作していない時にも流れてしまうリーク電流による消費電力である。静的な消費電力 P_s は式 (2) のように表すことができる。

$$P_s = N_t V I \quad (2)$$

(N_a :動作ノード数, C :静電容量, V :電源電圧,
 f :動作周波数, N_t :全ノード数, I :リーク電流)

両式を見ると電源電圧の影響が大きいことがわかる。しかし、現在CPUの電源電圧は約1Vであり、これはトランジスタの閾値に近い値であるため、これ以上上げると正常に動作しない可能性がある。そこで本研究では電源電圧よりも、主に動作周波数の変更による手法を考える。

3 P-state と C-state

プロセッサがアイドル状態になったとき、動作周波数の低減や、クロックの停止により消費電力を削減すること

ができる。このときプロセッサに用意されている P-state (Performance state) と C-state (CPU idle state) を用いて制御を行う。P-state は動作周波数や電圧を制御し、C-state は Halt 命令実行やクロック停止などのコアの休止状態を制御する。

1つのコアで処理が可能である場合、使わないコアは C-state により休止状態に変更される。一方、各コアが100%の処理スピードが必要無い時、P-state で動作周波数を下げて処理を行うことで電力削減ができる。P-state の適用よりも C-state の適用のほうが電力削減量は多いが、休止状態からの復帰に時間がかかる。

これら P-state、C-state の状態制御は Advanced Configuration and Power Interface (ACPI) 規格に従い、OS が各 state を選択する。

4 システム構成

本論文の評価では AMD 社のクアッドコア CPU であるサーバー向けの Opteron とデスクトップ PC 向けの Phenom を用いて評価を行った。これらの CPU の動作電圧は全てのコアで同じであるが、周波数は各コアで異なる値を設定可能である。つまり使っていないコアだけの電源を切ることにはできないが、動作周波数を低くしたり、0にすることは可能である。なお使用した OS は Windows Vista である。

4コア全てが100%の使用率で処理を実行する時と4コア全ての動作周波数が0Hzの時の消費電力を測定したところ、これらのプロセッサは約15~85Wの消費電力であることが分かった。

5 消費電力削減の手順

OSによる動的な電力管理を行うためにP-state、C-stateの設定とその動作についても考える必要があり、消費電力を構成する各要因ごとに考えていく。

5.1 静的な消費電力の削減

静的な消費電力はリーク電流により決まるが、式 (2) を見ると動作電圧を下げると消費電力も下がるように見える。しかし低電圧で動作させるためにはリーク電流の多いトランジスタを用いる必要があるため結果として電力

削減とはならない。

リーク電流はコア温度が高いほどその値も大きい。よって、動作周波数を下げることによってコア温度を下げ、リーク電流を削減することが期待できる。

5.2 動的な消費電力の削減

静的な消費電力の削減と同様に動作周波数を下げることによって実現する。その際ベンチマークソフトを用いて性能と電力削減のトレードオフを考える。

ベンチマークソフトには図 1 の評価では SPEC CPU2006 を、図 2、表 1 の評価では SYSmark2007 を使用した。

最小動作周波数の決定

まず、キャッシュコヒーレンスの影響による性能の損失と消費電力削減のトレードオフを考える。アイドル中のコアの周波数が低すぎると他のコアからキャッシュ内容の確認要求（プローブ）が発行された場合、動作周波数を上げてからプローブを実行する必要がある。その周波数変更時間が遅延となり、性能低下を生む。

周波数を変えずにプローブが行える周波数（ブレークオーバー周波数）を決定するためにアクティブなコアの周波数は最大に固定し、アイドルコアの周波数を変更して性能を測定した。その結果は図 1 のようになった。

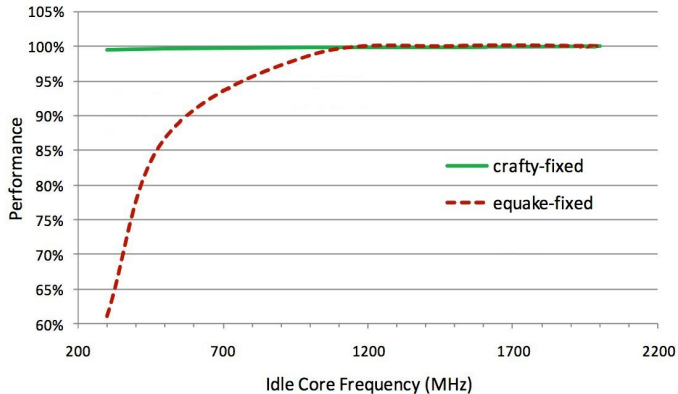


図 1: プローブに対するトレードオフ

図 1 によると約 1200MHz 以上の周波数であれば性能の損失が起きないことがわかった。よってアイドルコアの周波数は最低でも 1200MHz 以上とすることが望ましい。

P-state 変更時間間隔の決定

Windows Vista では、P-state は 10 ~ 100ms 間隔で負荷に応じた変更が行われる。変更時間間隔が短ければその時の負荷に応じた設定ができない可能性がある。そのため、設定可能な最短時間間隔である 10ms ごとに負荷の判断をさせ、OS 標準設定の結果とどれだけ性能の損失が異なるかを比較した。その結果が図 2 である。

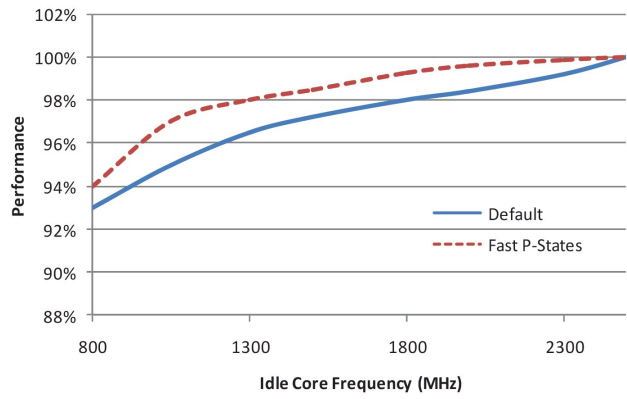


図 2: 変更間隔の違いによる性能の損失の比較

図 2 より性能の損失が少なくなったことがわかる。これより変更間隔時間が短いほどその状況にあったパラメータを OS が設定できると言える。

6 最適化の結果

変更間隔が OS 標準の設定 (Normal)、最短間隔の設定 (Fast)、最短間隔かつ性能重視のために動作周波数の変動幅が短い設定 (Fast-perf) の 3 種類について性能の損失と消費電力の比較を行った。その結果が以下の表 1 である。

表 1:最適化の結果

	P-States	Performance Loss	Power Savings
E-Learning	Normal	8.80%	43.10%
VideoCreation	Normal	6.20%	44.70%
Productivity	Normal	9.50%	45.30%
3D	Normal	5.90%	45.90%
E-Learning	Fast	6.40%	45.90%
VideoCreation	Fast	5.20%	46.10%
Productivity	Fast	8.00%	47.80%
3D	Fast	4.60%	48.20%
E-Learning	Fast-perf	1.50%	32.90%
VideoCreation	Fast-perf	1.80%	25.40%
Productivity	Fast-perf	2.50%	27.90%
3D	Fast-perf	1.40%	35.10%

Normal と Fast を比較すると Fast は性能の損失が少ないが、消費電力削減量は多いという結果となった。Fast-perf は性能の損失が平均 2 % にも関わらず平均 30 % の電力削減できた。

7 結論

Fast をバランス重視、Fast-perf を性能重視の設定とすることで、効果的に OS による省電力化が達成できると考えられる。

参考文献

- [1] Kim, N.S., Austin, T., Baauw, D., Mudge, T. Leakage current: Moore's law meets static power. IEEE Computer Society, pp.68-75, 2003