

消費電力の実行時予測を用いた電力最適化

高橋 英樹†

桑原 寛明†

國枝 義敏†

†立命館大学情報理工学部

1 はじめに

近年、PC クラスタの消費電力が大幅に増加している。消費電力の増加に伴う発熱の増加により、信頼性の低下や実装密度の低下、冷却コストの増加などのさまざまな問題が生じている [1]。

消費電力を抑える目的で、DVFS(Dynamic Voltage and Frequency Scaling) を利用する手法が数多く提案されている。DVFS はプロセッサの動作電圧と動作周波数を動的に変更する技術であり、さまざまなプロセッサに実装されている。

我々の研究グループでは、PC クラスタで扱う並列プログラムの実行には通信を伴うことが多く、通信時にプロセッサの動作周波数を下げれば処理時間を増大させずに省電力化を図れることに着目した [2]。しかし、通信のたびに DVFS 制御を実行するとオーバヘッドにより性能低下を起こす場合がある。そのため、プログラムをある程度の処理量の領域に分割し、領域ごとに DVFS 制御を行い、DVFS 制御の回数を減らす必要がある。また、領域ごとに最適な動作周波数を求める必要が出てくる。

本稿では、消費電力を動的に予測することにより各領域における最適な動作周波数をプログラム実行中に決定する手法を提案する。

2 関連研究

PC クラスタにおける DVFS を利用した電力最適化として、事前実行により得られたプロファイル結果を利用して最適な動作周波数を決定する手法 [3] がある。しかし、プログラムへの入力データの違いによりプログラム中の計算量や通信量が変化した場合、適切な DVFS 制御を行うことができず、プロファイリングをやり直す必要がある。

また、プログラムの実行中に、FLOPS 値を参照し最適な動作周波数を動的に決定し、DVFS 制御を行う手法 [4] がある。しかし、FLOPS 値を参照にする手法ではプロセッサの消費電力を考慮することはできるが、PC クラスタ全体の消費電力を考慮することができない。

3 消費電力の実行時予測と周波数の決定

本稿では、PC クラスタの消費電力を動的に予測することにより最適な動作周波数をプログラム実行中に決

定する手法を提案する。実際の実行時の処理時間と通信時間を用いて決定するため、入力データが違ってもプロファイリングをやり直す必要が無い。提案手法では、まずプログラムを複数の領域に分割し、領域ごとの処理時間と通信時間を用いて消費電力を予測し最適な動作周波数を決定する。

3.1 プログラムの領域分割

プログラムの領域分割は木村らの手法 [5] の考え方に基づく。以下にプログラム領域分割の手順を示す。

1. 基本ブロックに分割し、候補とする。
2. 各基本ブロックの境界に目印となるように printf 文で出力するコードを挿入し、1 度試行する。
3. 試行時の 100msec 毎の消費電力の値 [W] をプロファイリング結果として得る。
4. 3. のプロファイリング結果から、1. の基本ブロック分割において隣接する 2 ブロック間で、消費電力に変化がない場合と、処理と通信の比率が大きく変化しない場合には両者を 1 つの領域に統合する。

3.2 消費電力の予測

前節の領域分割により通常、ループはその前後で領域分割される。各領域に対して以下のように DVFS 制御を行う。

1. 事前に、選択可能な周波数 $F = \{f_1, \dots, f_n\}$ について、周波数 $f_i (i = 1, \dots, n)$ での計算時の単位時間当たりの消費電力 $E_{cal}(f_i)$ と、通信時の単位時間当たりの消費電力 $E_{com}(f_i)$ を測定しておく。
2. プログラムを実行するとともに、手法を適用するループの領域について繰り返しの最初の n 回を選択できる周波数 f_i すべてで実行し、そのときの計算に要した時間 $T_{cal}(f_i)$ と、通信に要した時間 $T_{com}(f_i)$ を測定する。
3. 以下の式でその領域を周波数 f_i で実行したとき予測消費電力 $E_{blo}(f_i)$ を求める。

$$E_{blo}(f_i) = E_{cal}(f_i)T_{cal}(f_i) + E_{com}(f_i)T_{com}(f_i)$$

4. ユーザの指定した許容される実行時間の増加率を δ とし、以下の式を評価する。

Power Optimization Using Dynamic Prediction on Power Consumption
Hideki Takahashi†, Hiroaki Kuwabara† and Yoshitoshi Kunieda†
†College of Information Science and Engineering, Ritsumeikan Univ.

$$T_{fast} = \min(T_{cal}(f_1)+T_{com}(f_1), \dots, T_{cal}(f_n)+T_{com}(f_n))$$

$$(\delta + 1)T_{fast} \geq T_{cal}(f_i) + T_{com}(f_i)$$

5. 4. の条件式を満たす周波数 f_i のうち, $E_{blo}(f_i)$ が最小となるものをその領域の最適動作周波数 f_{best} とし, $n+1$ 回目からは f_{best} で実行する.

4 評価

4.1 評価環境

評価実験では Intel Core 2 Duo E6750(2.66GHz, L2: 4MB) を搭載した 4 ノードクラスタを用いた. 設定可能な周波数は 2.66GHz と 1.98GHz の 2 種類である. カーネルは Linux kernel 2.6.22, コンパイラは gcc 4.1.2, MPI ライブラリは MPICH2 verison 1.0.5 を用いた. テストプログラムには NPB(NAS Parallel Benchmarks)3.2-MPI を用いた. 5 つあるカーネルベンチマークの中から IS について実験した. 今回実装したコードは C 言語のプログラムにのみ対応しているため, Fortran で実装されているほかの 4 つのカーネルベンチマークについての評価は除外した. 電力の測定にはクランプオンハイパワーテスタ 3169[6] を用いた.

4.2 評価実験

動的に動作周波数を決定することで, プログラムへの入力データが異なる場合でもプロファイリングのやり直しをせずに適切な DVFS 制御を行えることが, 提案手法の特徴である. それを示すために, IS を CLASS=C で試行した際のプロファイリング結果を用いてプログラム領域分割を行い, 各領域に対して提案手法により動的に動作周波数を決定し DVFS 制御を行った場合と, 静的に動作周波数を決定し DVFS 制御を行った場合(以下, 静的手法)について, IS を CLASS=C と CLASS=B で実行して実行時間と消費電力を比較する. また, 3.2 節で述べた許容する実行時間の増加率 δ はユーザが指定できるが, ここでは 0.2 とした.

表 1 に CLASS=C の場合について, プログラム全体を最高周波数で実行した場合に対する実行時間と消費電力の変化率を示す. また, 表 2 に CLASS=B の場合について, プログラム全体を最高周波数で実行した場合に対する実行時間と消費電力の変化率を示す. 表 1, 表 2 から共通して, 実行時間は少し増加しているものの, 消費電力が削減できていることが確認できる. しかしながら, 提案手法は静的手法と比べて実行時間の増加が大きく, 消費電力の削減量も少ない.

4.3 考察

提案手法は, プログラムへの入力データの違いにより領域内の処理量と通信量の比率が変わる場合に有効である. しかし, 表 1, 表 2 に示した結果から, IS においては入力データを変更した場合でも静的手法によ

表 1: CLASS=C での消費電力と実行時間の変化率

	提案手法	静的手法
実行時間	+2.87[%]	+1.34[%]
消費電力	-3.68[%]	-5.82[%]

表 2: CLASS=B での消費電力と実行時間の変化率

	提案手法	静的手法
実行時間	+1.48[%]	+0.52[%]
消費電力	-5.52[%]	-7.09[%]

る DVFS 制御が適切に動作していることが分かる. これは, 入力データの違いによって領域内の処理量と通信量の比率がほとんど変化しなかったためだと考えられる.

5 おわりに

本稿では, 電力プロファイルと実行時間を用いて予測した消費電力に基づく実行時最適動作周波数決定手法について述べた. 今後は, 今回除外した NPB のほかのカーネルベンチマークについて評価を行い, さらに検討を深める.

参考文献

- [1] Chung-Hsing Hsu and Ulrich Kremer : The Design, Implementation, and Evaluation of a Compiler Algorithm for CPU Energy Reduction, In Proc. of PLDI-2003, pp.38-48 (2003).
- [2] 高柳 圭孝: PC クラスタにおける DVFS を用いた並列プログラムの電力量削減のための基礎研究, 平成 19 年度 立命館大学修士学位論文 (2007).
- [3] 塚本 智博, 片桐 孝洋, 吉瀬 謙二, 弓場 敏嗣: 電力的制約を考慮した低消費電力指向最適化コンパイラ, 情報処理学会研究報告, 2007-ARC-172/2007-HPC-109, pp.55-60 (2007).
- [4] 木村 英明, 佐藤 三久, 今田 貫之, 堀田 義彦: 自動プログラム領域分割を用いた実行時 DVFS 制御, 情報処理学会研究報告, 2008-ARC-177/2008-HPC-114, pp.91-96 (2008).
- [5] 木村 英明, 佐藤 三久, 堀田 義彦, 今田 貫之: 影響の少ないインスツルメント手法と電力最適化のためのプログラム領域分割, 情報処理学会論文誌, Vol.48 No.SIG13(ACS 19), pp.247-259 (2007).
- [6] 日置電機株式会社: クランプオンハイパワーテスタ 3169, <http://hioki.jp/3169/index.html>