

Minimizing Energy Consumption for Precedence-constrained Applications Using Dynamic Voltage Scaling

著者： Young Choon Lee and Albert Y. Zomaya
 出典： 9th IEEE/ACM International Symposium on Cluster Computing and the Grid 2009, pages 92-99.
 発表者： 本多研究室 0953019 山下 良

1 はじめに

近年の環境意識の高まりや、電力需要の増大により、省エネルギー化への要求が高まっている。これまでの研究でも消費電力を考慮したスケジューリングの研究 [1] はあった。しかしながら、動作電圧の異なる CPU のヘテロジニアス (異種混在) 環境におけるスケジューリングの研究はこれまで行われていなかった。

そこで、本研究では、ヘテロジニアス環境を対象として CPU の電圧制御機構を使用した省エネルギースケジューリング手法を提案する。また、省電力化と性能はほとんどの場合において反比例するが、今回の提案ではこの点の考慮についても述べる。

2 消費エネルギー削減方法

2.1 対象とする環境

今回の研究では以下のような環境を対象としている。
 ・それぞれの CPU の最高動作電圧と処理性能が異なるヘテロジニアス (異種混在) 環境
 ・タスクの先行制約があるアプリケーション

2.2 電力の削減手段

研究の対象としているのは先行制約ありのアプリケーションであるため、各タスクを実行する前に先行制約タスクの処理が完了している必要がある。タスクを早く完了しても、他の先行制約タスクの処理完了待ちのため、次の処理へ移ることができない場合がある。

そのような場合、電圧を落として、他のタスクの処理完了時間とタイミングを合わせることができれば、電圧を落としている間は消費電力を低減できる。

消費電力は式 1 で示し、 n 個のタスクを処理した場合の消費エネルギー (電力量) を式 2 に示す。

$$P_c = ACV^2 f \quad (1)$$

A : CPU のスイッチの数 C : コンデンサの静電容量
 V : 動作電圧 f : 動作周波数

$$E = \sum_{i=1}^n ACV_i^2 f w_i \quad (2)$$

n_i : タスク番号 i のタスク

w_i : タスク処理完了までにかかる時間

これらの式から、電圧 V が消費電力と消費エネルギーに多大な影響を及ぼすパラメータとなっていることがわかり、動作電圧の制御で消費電力を低減できることがわかる。しかしながら、消費電力の低減のためにあまりに時間をかけすぎると、タスクの処理完了までの時間 (w_i) が長くなり、結果として消費エネルギーが増えてしまう場合があることを注意しなくてはならない。

3 スケジューリングのアルゴリズム

今回提案の省エネルギースケジューリングは以下の手順で実行する。

1. b-level を計算してリストとしてタスクを並び替える
2. RS(Relative Superiority) 指標を使って消費エネル

ギーの下がる電圧パターンをすべて計算する

3. MCER(Makespan conservative energy reduction) でスケジュール全体の完了時間や消費エネルギーの増減を計算し、消費エネルギーが削減でき、かつスケジュール長が伸びないパターンを選び出す。

4. 実行

3.1 b-level

今回は図 1 に示すタスクグラフを例にとって説明する。図 1 で円でかこまれた数字はタスク番号である。また、矢印中に挿入されている数字は通信時間を示している。

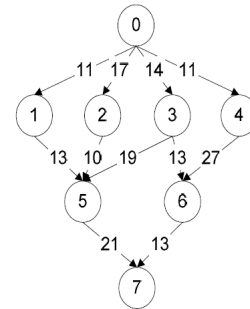


図 1: 例として用いるタスクグラフ

まずはじめに各タスクの各 CPU での処理時間を平均したものを算出する。算出した結果が表 1 である。

表 1: 平均処理時間

Task	平均処理時間(単位時間)
0	11
1	12
2	11.66
3	12.33
4	15
5	8.66
6	12.33
7	12

算出した値を元に各タスクにおいて、そのタスクまでもっとも長くかかる経路での到達時間を計算する。タスク 1 の場合はタスク 0 の処理時間 11 と通信時間 11 の合計 22 が到達時間である。同様に各タスクについて算出すると表 2 のように到達時間を算出できる。

この到達時間に最後のタスクの処理にかかる時間まで合算して、最後のタスクまでの終了時間の残りとして表したものが表 2 の b-level である。これを元に優先度リストを作成する。

3.2 RS 指標

消費電力低減とそれに伴う処理時間の変化を評価するための指標として RS 指標を用いる。RS 指標は以下の式

表 2: b-level と各タスクまでの到達時間

Task	b-level	到達時間
0	101.33	0.00
1	66.67	22.00
2	63.33	28.00
3	73.00	25.00
4	79.33	22.00
5	41.67	56.33
6	37.33	64.00
7	12.00	89.33

にて定義する。

$$RS(n_i, P_j, v_{j,k}, P', v') = -\left(\frac{E(n_i.p_j, v_{j,k}) - E(n_i, P', v')}{E(n_i.p_j, v_{j,k})}\right) + \left(\frac{EFT(n_i.p_j, v_{j,k}) - EFT(n_i, P', v')}{EFT(n_i.p_j, v_{j,k}) - \min(EST(n_i.p_j, v_{j,k}), EST(n_i.p', v'))}\right)$$

EST:Earliest Start Time EFT:Earliest Finish Time
 n_i : タスク番号 i のタスク P_j : j 番目のプロセッサ
 $v_{j,k}$: プロセッサ j における電圧 P' : 現在の消費電力
 v' : 現在の電圧

ここで前項は電力の削減量を示しており、後項は電圧などによって変化する処理時間を表している。この式で示される RS が大きくなるように電圧を考慮することで、スケジュール長があまり長くならず電圧を下げる事が可能となる。ここで電圧 V のパターンを何パターンも用意しておく。

3.3 MCER

先に述べた RS 指標では、一個のタスクにおいてしか最適な電圧を求めることができない。また、総処理時間についての考慮は不十分である。そこで、 RS 指標で暫定的に電圧 V を決定したあと、 $MCER$ で総消費電力と総処理時間を計算して、全体的なチェックを行う。この段階でスケジュール長が短くなっており、なおかつ消費エネルギーが低減できていれば、処理が実行されるが、そうでない場合は RS がもっとも大きくなる場合のスケジュールが採用され、実行される。

4 実験

評価はラプラス求解、高速フーリエ変換、LU 分解の 3 アプリケーションで実験を行った。比較対象は既存手法である、HEFT[2] と DBUS[3] である。実験のシミュレーション条件を表 3 に示す。

表 3: シミュレーション条件

Parameter	Value
The number of tasks	U(10, 600)
CCR	{0.1, 0.5, 1.0, 5.0, 10.0}
The number of processors	{2, 4, 8, 16, 32, 64}
Processor heterogeneity	{100, 200, random}

タスクの数は 10 個 ~ 600 個まで変化させる。表 3 に示されている CCR(Communication to Computation Ratio) は通信時間と計算時間の比である。通信時間が長くなると、今回提案のスケジューリング手法はスケジューリングに余裕ができ、既存手法より省エネルギーの点で有利になる傾向にある。しかしながら、あまりに長くなると、今回提案の手法でも消費エネルギーが大きくなる可能性があるため指標として用いている。

また、Processor heterogeneity はプロセッサのヘテロニアス度で一番遅いプロセッサと一番早いプロセッサの比 (%) である

5 結果

以下にラプラス求解での結果を示す (図 2, 図 3)

なお、ECR(Energy Consumption Ratio) はクリティカルパス法のエネルギー消費を 1 とした場合の各手法との比である。また、SLR(Schedule Length Ratio) はクリティカルパスのスケジュール長を 1 とした場合の各手法との比である。

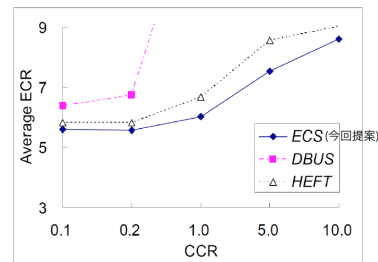


図 2: ECR の既存手法との比較

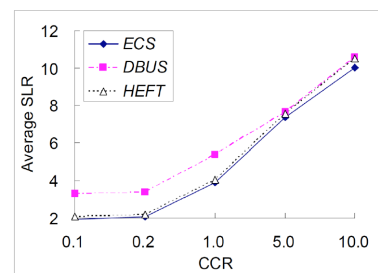


図 3: SLR の既存手法との比較

今回提案した手法である ECS(Energy Consumption Scheduling) が消費電力の点でも、スケジュール長の点でも優れている。ラプラス求解以外の結果については表 4 に示す。

表 4: Makespan(スケジュール長) と消費エネルギーの既存手法との比較

アルゴリズム ジョブの種類	DBUS		HEFT	
	Makespan	Energy	Makespan	Energy
Random	98%	60%	99%	88%
FFT	90%	59%	98%	91%
Laplace	77%	54%	95%	92%
LU	99%	76%	101%	90%
Average	91%	62%	99%	90%

6 おわりに

今回提案した低消費エネルギースケジューリング手法は非常に有効であった。これは $MCER$ によるものが大きい。また、CCR を変化させた場合も既存手法に比べて有利なことから、今回提案した手法はクラスシステムからグリッドシステムまで幅広い応用が見込まれる。

参考文献

- [1] Dusit Niyato et al. "Optimal Power Management for Server Farm to Support Green Computing", International Symposium on Cluster Computing and the Grid IEEE/ACM, pages 84-91, 2009.
- [2] H.Topcuoglu et.al "Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing", IEEE Trans. Parallel Dist. Systems, vol.10,no.8, pages 795-812, 1999.
- [3] Bozdog, D et al. "A task duplication based bottom-up scheduling algorithm for heterogeneous environments" Parallel and Distributed Processing Symposium, pages 132-143, 2006.