

マルチプロセッサにおける DVS を用いたリアルタイムタスクの消費電力の削減

三津橋諒也[†]横山孝典^{††}兪明連^{†††}東京都市大学[†]東京都市大学^{††}東京都市大学^{†††}

1. はじめに

近年の並列処理の技術進歩により、マルチプロセッサ・マルチコア環境が一般的になっている。しかし、処理能力の向上により消費電力も増加しており、その大部分はプロセッサによるものが占めている^[1]。プロセッサを複数使用するマルチプロセッサ環境において、プロセッサの消費電力を削減することは、全体の消費電力量を抑えるのに非常に有効であると考えられる^[2]。

プロセッサの消費電力を抑える技術には DVS (Dynamic Voltage Scaling)^[3]がある。DVS は動作電圧と動作周波数の組み合わせ P-State を動的に切り替えることにより消費電力と処理時間を変化させる。しかし、DVS はプロセッサの処理速度を低下させ消費電力を削減する技術であるため、高負荷時の利用はほとんど考慮されていない。

高負荷時の DVS 適用法については様々な方法が提案されているが、すべてのスケジューリングアルゴリズムに対応した適用法はいまだ確立されていない。そこで、一般的なスケジューリングアルゴリズムに対応した DVS 適用アルゴリズムを提案する。

2. 従来研究

Kimura らはタスクが実行されていない処理待ち時間があるタスクに対して DVS を行うことにより消費電力の削減に貢献している^[4]。しかしプロセッサ間の通信コストが発生する環境下では適切に DVS を適用できないという問題点が発生する。

森らは、ETF (Earliest Task First)^[5]アルゴリズムを対象とし、後続タスクも考慮したタスクの実行が遅延しても全体の実行終了時間が増加しない時間(以下 slack-time)を算出し、その時間内で DVS を適用することにより、プロセッサ間の通信コストが発生する場合に対応し、消費電力削減にも貢献している^[6]。しかし、時間的制約のあるタスクについて考慮されておらず、デッドラインのあるタスクに適用した場合デッドラインミスが発生する可能性がある。

3. 提案手法

3.1 提案アルゴリズム

デッドラインを持つタスクにも対応するため、slack-time の算出条件に「タスクの実行が遅延しても自身と後続タスクのデッドラインミスが発生しない範囲」を加えることによって、ETF アルゴリズムだけではなくデッドラインを持つアルゴリズムに DVS を適用した場合においてもデッドラインミスの発生を増加させずに消費電力を削減することができる。

表 1. 計算機環境

構成要素	仕様
マザーボード	Gigabyte GA-K8 VM800M
CPU	AMD Athlon64 3000+
クロック周波数	2.00GHz
P-state の数	3

Power reduction of real time task with DVS on multi-processor

[†] Mitsuhashi Ryoya · Tokyo City University

^{††} Yokoyama Takanori · Tokyo City University

^{†††} Yoo Myungryun · Tokyo City University

表 2. 消費電力モデル

プロセッサのP-State			実測値		
P-State	周波数 [MHz]	電圧 [V]	処理時間の比	消費電力[W]	
				アイドル	高負荷
1	2000	1.50	1.0	87	100
2	1800	1.40	1.1	78	87
3	1000	1.10	2.0	59	61

表 1, 表 2 に示す計算機環境と消費電力モデルに基づき, ETF スケジューリングアルゴリズムにより生成されたスケジュールを入力とし提案アルゴリズムで DVS を適用する.

3.2 適用例

ETF スケジューリングアルゴリズムによって生成されたスケジュールを図 1 に示す. 各タスクはデッドラインを持っていないため, デッドラインを考慮する必要はない. slack-time は全体の実行時間が増加しない範囲で DVS を適用できる時間であるため, タスク 2, タスク 4 以外のタスクは slack-time が 0 となり, DVS が適用できない.

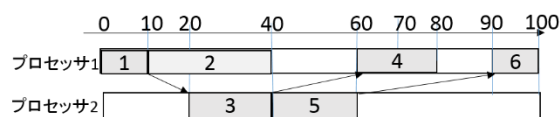


図 1. 入力スケジュール

図 1 のスケジュールに DVS を適用した結果を図 2 に示す. デッドラインのないタスクグラフであるため, 適用結果は森のアルゴリズムと同じになる

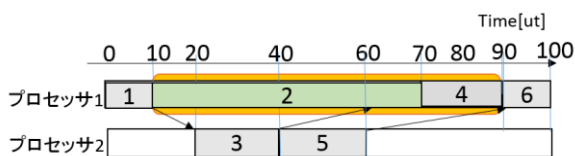


図 2. デッドラインなし

次に, 適用前のスケジュールのタスク 2 にデッドライン 70[ut], タスク 4 にデッドライン 80[ut] が設けられている場合に提案アルゴリズムによって DVS を適用した結果を図 3 に示す. タスク 2 は 70[ut] まで終了時間が遅延しても全体の実行終了時間は増加しない. しかし, タスク 2 に P-state3 で DVS を適用した場合, タスク 2 のデッドラインは守られているが, タスク 2 の実行終了が遅れるためタスク 4 にデッドラインミスが発生

してしまう. その結果タスク 2 の P-state は 2 となり, 他タスクは slack-time を持っていないため, DVS 適用処理を終了する. 適用可能タスクだけでなく, 後続タスクすべてのデッドラインを考慮する必要がある.

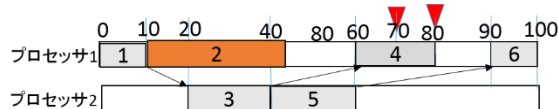


図 3. タスク 2 とタスク 4 にデッドライン

4. 結果及びまとめ

デッドラインを持つタスクを対象にし, プロセッサ 2, 4, 8 個の場合における消費電力削減量とデッドラインオーバー量の比較を行った. 従来研究ではデッドラインミスが発生する場合があったが, 提案手法ではデッドラインミスを増加させることなく消費電力を削減することができた.

今後の課題として, P-state を切り替えること自体のオーバーヘッドによって実行時間や消費電力量が増加している可能性があるため, P-state 切り替えによるオーバーヘッド量の調査が挙げられる.

謝辞

本研究の一部は JSPS 科研費 15K00084 の助成を受けたものです.

参考文献

[1]岩田淳, 高瀬英希, 高木一義, 高木直史, "組み込みシステム向け DVFS アルゴリズムのリアルタイム OS への実装および比較評価", FIT2013, pp. 315-318 (2013)
 [2]L. A. Barroso, "The Price of Performance", Queue, Vol. 3, No. 7, pp. 48-53 (2005)
 [3]Y. Zhang, X. S. Hu, and D. Z. Chen, "Task Scheduling and Voltage Selection for Energy", Proc. Of the 39th annual Design Automation Conf., PP. 183-188 (2002)
 [4]H. Kimura, M. Sato, Y. Hotta, S. Matsuoka, T. Boku, and D. Takahashi, "Empirical Study on Reducing Energy of Parallel Progr ce, Power-Aware Computing, pp. 231-250 (2005) ams using Slack Reclamation by DVFS in a Power scalable High Performance Cluster", Proc. Of 8th IEEE Int'l. Conf. on Cluster Computing (CLUSTER), CD-ROM (2006)
 [5]森雄一朗, 朝倉宏一, 渡邊豊秀, "マルチプロセッサ環境における DVS を用いた消費電力量削減アルゴリズムの構築", 電学論 C, 131 巻 4 号, pp. 926-933, 2011