Vol.2013-ARC-206 No.22
2013/8/1

DVFSドメインを考慮した 低消費電力化タスクスケジューリング手法に関する検討

和田 康孝[1,a)]　近藤 正章[1]　本多 弘樹[1]

概要：コンピュータシステムの消費電力を低減するために現在では様々な技術が用いられており，特にプロセッサのDVFS（Dynamic Voltage-Frequency Scaling）機能の活用は，パワーゲーティングの適用とともに，プログラム実行時の消費電力および消費エネルギーを削減するために有効な手段である．従来のDVFSスケジューリング手法では，各コア個別にDVFSを適用可能であるという条件のもと，プログラムの実行時間の増加を抑えつつ消費エネルギーを削減可能であることを示しているが，将来的にはチップ上のコア数が増大し，各コア毎にDVFSを適用することは難しくなるものと考えられる．本稿では，メニーコアプロセッサにおいて，複数コア単位でDVFSが適用可能である場合を考慮した低消費電力化タスクスケジューリング手法について検討・評価を行った結果について述べる．

1. はじめに

半導体集積技術の進歩に伴い，従来はHPCシステムやハイエンドPCに用いられていたマルチコアプロセッサが組込みシステムにおいても用いられるようになってきている．これは，組込みシステムにおいても処理性能への要求が高まってきているためであるが，低消費電力化への要求は依然として高く，性能の向上と消費電力の低減を両立する必要がある．さらに近年では，環境問題やシステム全体への電力供給の問題などから，HPCシステムにおいても消費電力が重要視され始めており，あらゆる計算機システムにおいて消費電力の削減が重要な課題であると言える．

一般に計算機システムにおいて最も消費電力の大きい要素はプロセッサであるため，消費電力・消費エネルギーを削減する手法としては，プロセッサ内部の回路への電源供給を遮断してリーク電力を含む消費電力を削減するパワーゲーティング（Power Gating，PG）や，プロセッサの動作周波数と電圧を適切に制御して消費電力を削減するDVFS（Dynamic Voltage-Frequency Scaling）がよく用いられる．マルチコアプロセッサ上で並列アプリケーションを実行する際にPGやDVFSを適用し，消費電力および消費エネルギーを削減する手法は従来から研究が進められており，例えば，自動並列化コンパイラにおいてプログラムの階層構造と内部のタスクの依存関係を考慮しDVFSとPGを適用する手法 [1], [2] や，並列プログラム内部の処理（タスク）間の依存関係とアプリケーション全体の実行時間の上限を考慮し，DVFSを適用する時間的余裕度を分配する手法 [3]，タスク間のデータ転送にかかる消費エネルギーを削減するようにタスクをクラスタリングしてスケジューリングを行いDVFSを適用する手法 [4]，処理能力の異なる複数のプロセッサを持つヘテロジニアスなシステムに対してアプリケーションの処理時間と消費エネルギーを考慮してタスク割当を行う手法 [5] などが提案されている．

これらの手法のほとんどは，ノード単位あるいはコア単位でDVFSやPGが適用可能であるモデルを前提にしているが，実際には，プロセッサに対する電圧の供給はチップ一括で行われている場合が多く [6]，これらのモデルをそのままマルチコア上で適用することはできない．近年では電圧レギュレータをチップやパッケージに統合して複数の電圧ドメインを個別に制御することも行われているが，チップ上のコア数が多いメニーコアの場合には，複数のコアをグループ化し，このグループ単位で電圧を供給することになる [7]．電圧ドメイン数の増加はコア数の増加と比較して緩やかであると考えられるため，動作周波数と電圧の制御単位が大きく異なる状況において効果的にDVFSを適用し，アプリケーション実行にかかる消費エネルギーを削減する手法が求められる．

本稿では，DVFSの適用単位が複数コア単位である場合を考慮した低消費電力化タスクスケジューリング手法について検討・評価を行った結果について述べる．

以下，2章で対象とするマルチコアおよび並列アプリケー

[1] 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems,
The University of Electro-Communications, Tokyo, Japan
[a)] wada@is.uec.ac.jp

ションのモデルについて，3 章で評価・検討を行う DVFS スケジューリング手法について，4 章で標準タスクグラフセット [8] を用いた性能評価について述べる．

2. 対象モデル

ここでは，本稿が対象とする DVFS 機能を持つマルチコアプロセッサのモデル，およびスケジューリング対象となる並列アプリケーションのモデルについて述べる．

2.1 DVFS スケジューリング

多くの場合，マルチコアをはじめとする並列計算機環境を対象とした DVFS スケジューリングは 2 つのステップからなる．1 つ目のステップは各コアやノードへのタスクスケジューリング，2 つ目はタスクスケジューリング結果に対する周波数・電圧制御である．手法によっては，消費電力・消費エネルギーを削減するために 2 つ目のステップでタスクの割り当てを変更することも考える．

また，DVFS は，処理時間を延ばす代償としてタスクあるいはアプリケーションの消費電力と消費エネルギーを削減することを可能とするものである．そのため，各アプリケーションに対して，許容できる実行時間の上限が与えられる．上記 2 つ目のステップでは，この上限を超えない範囲で DVFS や PG を適用して消費電力と消費エネルギーの削減を目指す．

2.2 対象マルチコアのモデル

1 章で述べた通り，将来的に 1 チップ上に集積されるプロセッサコア数が増加しメニーコア化が進んでいくことを考慮すると，動作周波数はチップ上のコア単位で制御することが可能であっても，コア単位で電圧ドメインを形成することは難しくなる．そのため，複数個のコアをグループとして DVFS ドメインを形成することになるものと予想される．それに対して，PG は従来からコア内の演算器単位など粒度の小さい単位でも適用されており，将来的にもコア単位での適用は可能であると考えられる．

以上の点を考慮して，本稿において対象とするマルチコアプロセッサのモデルを下記のように定める．

- パワーゲーティングはコア毎に個別に適用可能
- 動作周波数はコア毎に動的に制御可能
- 設定可能な動作周波数の状態は離散的であり，各動作周波数での動作に必要な電圧の値も定められている
- 電圧は 1 つ以上のコアが属するドメイン単位で設定可能
- DVFS ドメインの電圧は，ドメイン内で最も高い動作周波数のコアにあわせて，システムによって動的かつ自動的に制御される

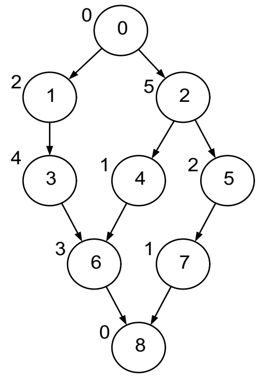

図 1 タスクグラフの例

2.3 並列アプリケーションのモデル

本稿において検討するスケジューリングアルゴリズムでは，粗粒度タスク並列性を持つアプリケーションを対象としており，アプリケーション内の粗粒度タスクおよびタスク間の依存関係は有向非循環グラフによって表現される．つまり，n 個の粗粒度タスクを表すノード集合 N およびタスク間の依存関係を表すノード間の有向エッジ集合 E からなるタスクグラフ $G = (N, E)$ として記述される．なお，このタスクグラフは 1 つの入口ノードと 1 つの出口ノードを持ち，全てのノードは入口ノードおよび出口ノードから到達可能である．この入口ノードおよび出口ノードは実行コスト 0 のダミーノードであり，一般性を失わずにこの形式を適用可能である．図 1 に対象とするタスクグラフの例を示す．図 1 において，各ノードはアプリケーション内の 1 つの粗粒度タスクを表し，ノード内の数字はタスク番号 i，ノード左上の数字は当該タスクの実行コスト t_i を表す．また，ノード i からノード j へのエッジは，タスク j はタスク i が完了しなければ実行を開始できないという制約を表している．本稿で扱うスケジューリングアルゴリズムにおいては，これらの制約を満たすように各コアにスケジューリングする必要がある．このとき，各タスクはノンプリエンプティブに実行されるものとする．

2.4 対象モデルにおける DVFS

ここでは，上記のマルチコアモデルおよびアプリケーションモデルにおいて，DVFS および PG がどのように適用されるかについて述べる．図 2 はコア 0 およびコア 1 が 1 つの DVFS ドメインを形成しており，コア 0 にタスク i および j が，コア 1 にタスク k および l が割り当てられている例である．図 2 において，タスク i および l は動作周波数 MID で，タスク k は FULL で，タスク j は LOW でそれぞれ実行されている（ただし，動作周波数は $FULL > MID > LOW$）．このとき，図中の時刻 $T1$ まではコア 0 の動作周波数が MID，コア 1 がの動作周波数が FULL であるため，ドメイン全体の電圧はコア 1 を動作させるために必要なレベルにあわせて設定される．時刻 $T1$

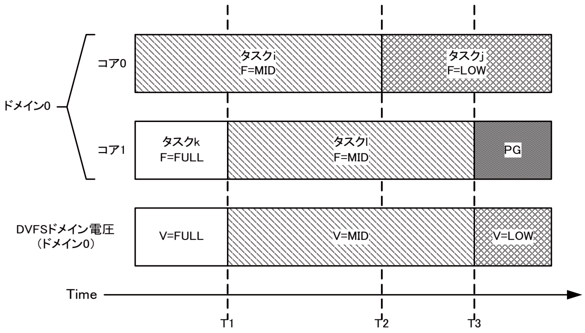

図 2　DVFS・PG の適用例

から $T2$ の間はコア 0・1 とも動作周波数 MID でタスクを実行しているため，ドメインの電圧もこれにあわせて時刻 $T1$ で MID に下げられる．$T2$ から $T3$ の間はコア 0 の動作周波数が LOW に制御されるが，コア 1 が引き続き動作周波数 MID でタスクを実行しているため，ドメインの電圧は変更されない．時刻 $T3$ 以降はコア 1 に PG が適用されるため，コア 0 の動作周波数にあわせて電圧が LOW に設定される．以上のように，PG および動作周波数制御はコア単位で適用されるが，電圧制御は DVFS ドメイン内の最も動作周波数の高いコアにあわせて行われる．また，動作周波数の変更は粗粒度タスクの実行開始あるいは終了時に行うこととし，タスクの途中では行われないものとする．ただし，電圧制御はドメインに所属するコアの動作周波数に合わせて動的に行われるため，この限りではない．図 1 に例示するタスクグラフ内の各タスクにおいて，タスク開始から終了までの消費電力は一様であり，動的電力は動作周波数および電圧の 2 乗に比例し，各タスクの実行時間は動作周波数に反比例するものとする．

なお，本稿においては，粗粒度タスクのコストは十分大きいものとし，タスク間の通信や同期，低消費電力制御に関わるオーバヘッドは無視できるものとする．

3. DVFS スケジューリングアルゴリズム

ここでは，本稿において評価・検討を行う DVFS スケジューリングアルゴリズムについて述べる．2 章で述べた実行モデルにおいて効率よく DVFS を適用し，アプリケーション実行のために消費されるエネルギーを削減するには，下記の点を考慮して各粗粒度タスクをスケジューリングする必要がある．

- 同じ DVFS ドメイン内のコアがなるべく同じタイミングで同じ動作周波数に制御される
- 粗粒度タスクが連続して同じコアに割り当てられることを避け，動作周波数を下げる余地を確保する

本稿では，これらの点を考慮し，基準となる DVFS スケジューリング手法を含めた下記 4 種類の手法について評価を行った．自動並列化コンパイラなどの並列化支援ツールやランタイム等によってタスクスケジューリングを適用することを考えると，最適化アルゴリズムはスケジューリングにかかる時間が長大になってしまい，適当であるとは言い難い．そのため，本稿ではリストスケジューリングをベースとした手法について検討を行う．

3.1 ベースライン

ここでは，本稿における性能比較の基準，つまりベースラインとなる DVFS スケジューリングアルゴリズムについて述べる．この手法は Wang らが提案した手法 [5] を参考に，対象モデルへの適用が可能となるように改変を加えたもので，下記のステップにより実現される．

Step 1： CP/MISP 法 [9] により粗粒度タスクの割当先と実行順序を決定する．この時点では，全てのタスクの設定動作周波数は最速に設定される．

Step 2： アプリケーション実行時間の上限を超えない範囲で，タスクグラフ内の最長パス（Critical Path, CP）上にある粗粒度タスクの動作周波数を下げる．以降，CP 上に存在するタスクの設定動作周波数は変更されない．

Step 3： 目標動作周波数を最速の動作周波数とする．

Step 4： 現在の目標動作周波数が選択可能な最遅のものであれば処理を終了，そうでなければ，目標動作周波数を一段階遅いものに変更する．

Step 5： CP 上にはない全てのタスクに対し，タスクの割当順序，依存関係，およびアプリケーション実行時間の上限をもとに最早開始時刻および最遅完了時刻を求め，各タスクに対して設定可能な動作周波数を求める．

Step 6： 設定動作周波数を目標動作周波数に変更する対象のタスクを選択する．この際，(1) 現時点での設定動作周波数が目標動作周波数より高く (2) 目標動作周波数に設定可能なタスクのうち，最もタスク実行開始時刻が早いものを対象タスクとして選択する．

Step 7： 対象タスクが存在すれば，対象タスクの設定動作周波数を目標動作周波数に変更し Step 5 へ．対象となるタスクが存在しない場合，Step 4 へ．

この手法自体は DVFS ドメインを考慮したものではないため，タスク毎の動作周波数を下げることができたとしても，DVFS ドメインの電圧を下げることにはつながらない場合があり，アプリケーション実行にかかる消費エネルギーを削減するには課題が残る．

3.2 サイクリックなタスク割当

CP/MISF 法はリストスケジューリングの一種であるため，ナイーブな実装では，当該スケジューリング時刻においてタスクを実行していないコアのうち，最も番号の若い

ものを割当先として選択する．そのため，常にタスクを実行しているコアと，アイドル時間の割合が大きいコアの間で大きなばらつきが存在する．後者は割り当てられたタスク間に時間的な余裕があり，タスクの動作周波数を下げる機会を確保しやすいと考えられるが，前者はアイドルとなっている時間が短いため，タスクの動作周波数を下げる機会が少ないといえる．

このような特性を考慮して，この手法においては，CP/MISF 法においてタスクの割当先コアを選択する際，DVFS ドメインをサイクリックに巡回して探索を行う．つまり，あるタスクを DVFS ドメイン n に属するコアに割り当てた場合，次のタスクの割当先は DVFS ドメイン $(n+1)$ 内のコアから探索を開始する．

上記の方法に基づいてタスクのスケジューリングを行った後，ベースラインアルゴリズムの Step 2 以降と同様に実行時間の上限を考慮して各タスクを実行する際の動作周波数を決定する．

3.3 最遅終了時刻と最早開始時刻の差分を考慮したスケジューリング

ベースライン手法でも動作周波数決定のために用いられているように，タスク間の依存関係とアプリケーション実行時間の上限等から，各タスクの最早開始時刻と最遅終了時刻を求めることができる．これら 2 つの時刻の差から当該タスクの最低動作周波数を予想可能である．この手法では，同時期に実行開始されるタスクのうち，この予想最低動作周波数が同じタスクがなるべく同じ DVFS クラスタのコアに割り当てられるようにスケジューリングを行う．こうすることで，DVFS ドメインに含まれるコアの動作周波数が一致する可能性が高まり，効果的に DVFS が適用されることが期待できる．具体的には，CP/MISF 法でタスクを割り当てる際に，割当先のコアを下記の優先度に基づいて選択する．

1. 当該スケジューリング時刻における予想動作周波数が，現在割当を行っているタスクのそれと同じドメインから空いているコアを選択
2. 当該スケジューリング時刻において，全てのコアがアイドルであるドメインから空いているコアを選択
3. 当該スケジューリング時刻における予想動作周波数が，現在割当を行っているタスクのそれより大きいドメインから空いているコアを選択

ここで，DVFS ドメインの予想動作周波数とは，あるスケジューリング時刻において当該ドメインに含まれるコアに割り当てられているタスクの予想最低動作周波数のうち最も高いものをいう．

図 3 に最遅終了時刻と最早開始時刻を考慮したタスク割当の例を示す．図 3 において，コア 0 および 1 が 1 つの DVFS ドメインを，コア 2 および 3 が別のドメインを形成

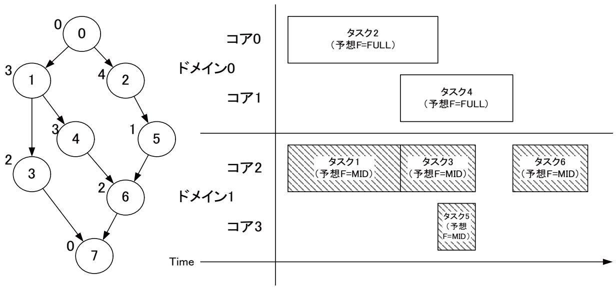

図 3 最遅終了時刻と最早開始時刻の差分を考慮した割り当ての例

している．初期状態で実行開始可能なタスクはタスク 1 および 2 であり，まず CP 長の大きいタスク 2 について割当を考える．この時点ではまだほかに割り当てられているタスクが存在しないため，コア 0 へ割り当てる．このとき，ドメイン 0 の予想動作周波数は FULL となる．次にタスク 1 の割当先を決定する際，タスク 1 の予想動作周波数は MID であるため，まだタスクの割当が行われていないドメイン 1 からコア 2 を選択する．次に実行開始可能なタスクが現れるのはタスク 1 の終了時刻であり，タスク 4 および 3 が対象となる．タスク 4 およびタスク 3 の予想動作周波数はそれぞれ FULL，MID であるため，まずタスク 4 はこの時点で予想動作周波数が FULL であるドメイン 0 から割当先を探索し，コア 1 を選択する．コア 3 はこの時点で実行中のタスクが存在しないドメイン 1 から割当先を選択する．以上を繰り返し，ダミーノードを除くグラフ中の全てのタスクについて割当を決定する．

このようにしてタスクスケジューリングを行った後，3.1 節のベースラインアルゴリズム Step 2 以降と同様に各タスクの動作周波数を実際に決定する．

3.4 スレッド割当の最適化

同じ DVFS ドメイン内のコアがなるべく同じタイミングで同じ動作周波数に制御されている機会をなるべく多く確保するためには，動作周波数の変化の傾向が近いスレッド同士を同じドメインに割り当て，そうではないスレッドを異なるドメインに割り当てることができれば，ドメインの電圧を下げる機会を増やすことができる．ここで，スレッドはある 1 つのコアに割り当てられたタスク群のことをいい，各タスクの開始・終了時刻と実行の順番，動作周波数の情報を含む．

この手法では，各スレッド間の距離を算出し，DVFS ドメイン内のスレッド間距離の合計を最適化する．スレッド間の距離は各時点でのスレッドの動作周波数状態の差分を積算したものであり，例えば，図 2 において，時刻 $T1$ まではコア 0 が実行するスレッドが周波数 FULL，コア 1 が周波数 MID で 1 段階異なるため，単位時間あたりの距離を 1 として積算する．時刻 $T1$ から $T2$ までの間は両方とも同じ動作周波数であるため，距離は 0 である．時刻 $T3$

表 1 各動作状態における周波数と電圧（相対値）

状態名	周波数の比	電圧の比
FULL	1.0	1.0
MID	0.5	0.85
LOW	0.25	0.70
OFF	0	N/A

以降はコア 1 に PG が適用されるため，この場合も距離は 0 として換算する．このようにして，アプリケーションの開始から終了までスレッド間の距離を積算し，動作周波数の推移の相違度を求める．その後，スレッドの入れ替えによってドメイン内スレッド間の距離，つまり相違度の合計が小さくなるようにスレッドとコアの対応付けの入れ替えを順次行う．

この手法はこれまでに述べた 3 種類の手法とは異なり，タスクスケジューリングと動作周波数の決定を行った後に適用される．つまり，上記の各手法と併用することが可能である．

4. 性能評価

3 章で述べた DVFS スケジューリングアルゴリズムについて消費エネルギー削減効果の評価を行った．

4.1 評価条件

本評価では，入力タスクグラフとして，標準タスクグラフセット [8] からタスク数 500（入口ノード・出口ノードを含めた場合，タスク数 502）の 180 例を用いた．各アプリケーションの実行時間の上限，つまり実行時間の余裕度としては，各タスクグラフの出口ノードから入口ノードまでの最長パス長に対して 0%から 100%までの間で変化をさせて評価を行った．ただし，DVFS 適用前の時点でこの制約を違反している場合は，DVFS 適用前のスケジューリング長を実行時間の上限とした．

また，比較的コア数の多い状況を想定し，マルチコア内の総コア数が 8，16 および 32 の場合について，DVFS ドメイン数を 1，2，4，8 と変化させて評価を行った．設定可能な動作周波数および電圧の状態は FULL，MID，LOW および PG 適用時を示す OFF の 4 種類とした．それぞれの状態における相対的な動作周波数および電圧を表 1 に示す．各コアが動作している際のリーク電力は，最大動作周波数（FULL）で動作している際のダイナミック電力を基準とし，その 10[%] で一定であると想定した．

4.2 評価結果と考察

アプリケーション実行時間に対する余裕度をタスクグラフ CP 長の 0%，30%，50%，80%，100%としたときの評価結果をそれぞれ図 4，5，6，7，8 に示す．本評価では，CP/MISF 法によりタスクスケジューリングを行った

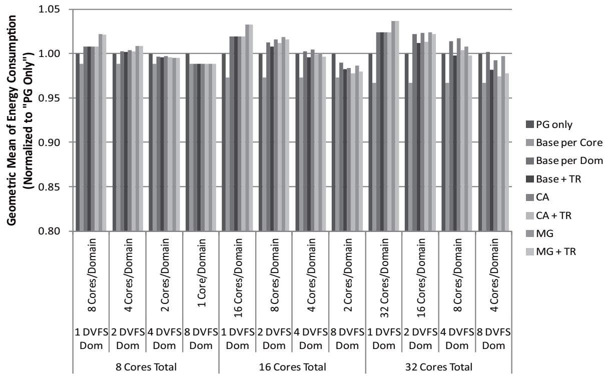

図 4 余裕度 0[%] における評価結果

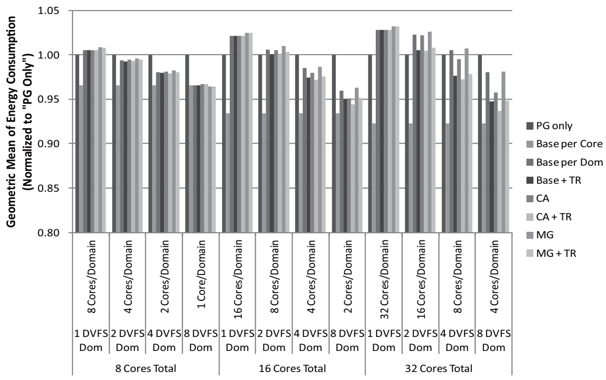

図 5 余裕度 30[%] における評価結果

後 PG のみを適用した場合（PG Only）を基準とし，コア毎に DVFS が適用可能である条件においてベースライン手法を適用した場合（Base per Core），ベースライン手法のみを適用した場合（Base per Dom），ベースライン手法に加えスレッド配置最適化を適用した場合（Base + TR），サイクリックなタスク割当を行った場合（CA），サイクリックなタスク割当とスレッド配置最適化を併用した場合（CA + TR），タスクの最早開始時刻・最遅終了時刻を考慮したスケジューリングを行った場合（MG），タスクの最早開始時刻・最遅終了時刻を考慮したスケジューリングとスレッド配置最適化を併用した場合（MG + TR）について評価を行った．

また，各図において，縦軸は各タスクグラフに対する消費エネルギーの相乗平均を “PG Only” を基準に正規化した値を示しており，横軸はコアの数および DVFS ドメイン数の組み合わせを表す．

図 4，5，および 6 を見ると，アプリケーション実行時間の余裕度が比較的小さく，かつ DVFS ドメイン数が小さい場合，DVFS を適用することによって却って消費エネルギーが増加する場合が見られる．これは，コアの動作周波数を下げても DVFS ドメインの電圧が下がらず，ダイナミック電力による消費エネルギー削減できない場面が多いためである．この場合，粗粒度タスク実行時間が延びてスタティック電力をより長く消費し続けてしまうため，か

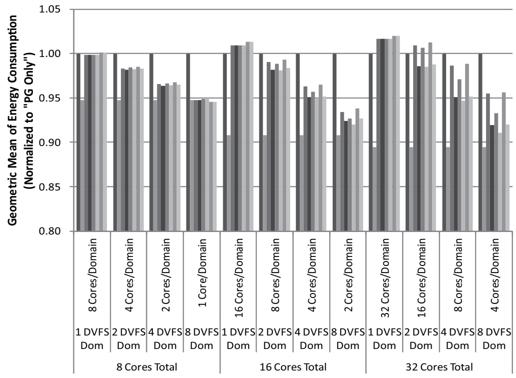

図 6　余裕度 50[%] における評価結果

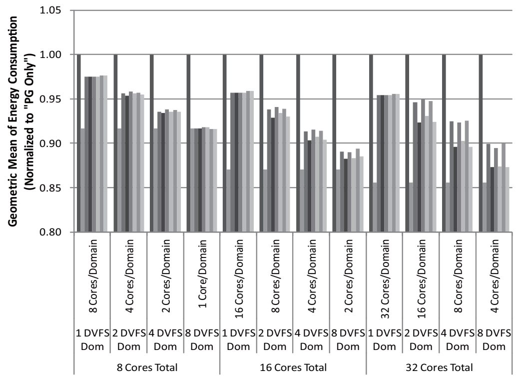

図 7　余裕度 80[%] における評価結果

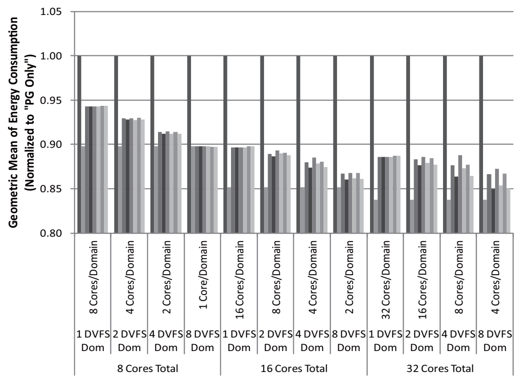

図 8　余裕度 100[%] における評価結果

えって消費エネルギーの増大を招く．この問題を解決するためには，DVFS ドメインの電圧がタスクの動作周波数に対して高く設定されている場合，タスクの動作周波数もそれにあわせて変更するという方法が考えられるが，依存関係にあるタスクへや消費電力への影響も考慮する必要があり，今後の課題である．

また，余裕度が 50%より小さい場合に DVFS の効果が現れにくいのは，選択可能な周波数が離散的であることが主な原因であると考えられる．

評価結果から，最遅終了時刻と最早開始時刻の差分を考慮したスケジューリング手法は，それ単体では効果が高くないことがわかる．これは，最遅終了時刻と最早開始時刻から予想される動作周波数が最終的に決定されたものとは異なることが多いためである．周波数制御によってタスク間の関係がかわる可能性があることも考慮して予測の精度を上げることが課題として挙げられる．

全体で 8 コアのシステムを想定した場合，各手法で大きな差が見られないことがわかる．これは，入力となるタスクグラフの並列度（並列度=コストの総和/CP 長）と比較してコア数が少ないため，コアがアイドルになる機会，つまり DVFS の適用機会がスケジューリング手法によって大きく変わらないためである．全てのコアが同じ DVFS ドメインに属している場合も同様に手法間の性能差が見られないが，これは，ドメイン内に 1 つでも高い動作周波数でタスクを処理しているコアがある場合は電圧もそれにあわせて高い状態に設定されてしまうためであると考えられる．この場合，消費エネルギーの削減はもっぱら PG に頼ることになる．

全体のコア数が多い 32 コアの場合に注目すると，スレッド配置最適化の効果が 16 コアの場合と比較して高いことがわかる．これは，タスクグラフの並列度と比較して全体のコア数が多くなる場面が増えるためであると考えられる．つまり，スレッド配置最適化によって DVFS ドメイン内で同時に実行されるタスクの数が減り，より低い電圧に設定される機会が大幅に増えるためである．

16 コアの場合，特に余裕度の小さい場合においてサイクリックなタスク割当の効果が高い．スレッド割当の最適化と併用することで，DVFS ドメイン数 8（ドメインあたり 2 コア）の場合，コア毎に DVFS が適用可能である場合（Base per Core）と比較して 1[%] 程度の消費エネルギー増加に押さえることができている．スレッド割当の最適化はいずれのコア数でも効果が高く，有効な手法であると言える．

5. まとめ

本稿では，複数のグループ単位で DVFS が可能なマルチコア環境において，DVFS ドメインを考慮したタスクスケジューリング手法に関する検討と評価を行った．アイドルになる機会をなるべく均等に分配するサイクリックなタスク割当，各タスクの最早開始時刻と最遅完了時刻から予想される動作周波数を考慮してタスクの割当先を決定する手法，動作周波数の推移が類似しているスレッドを同じ DVFS ドメインのコアに割り当てるスレッド割当最適化手法について評価を行った結果，アプリケーション実行時間の余裕度が比較的小さい場合にサイクリックなタスク割当が有効であると確認できた．また，スレッド割当最適化はいずれの場合でも比較的効果が高く有効であることが確かめられた．

今後の課題として，タスク間のデータ転送や同期のオーバヘッドを考慮した手法の検討や，性能や特性の異なるコ

アを搭載したヘテロジニアスマルチコアに対する適用などが挙げられる.

謝辞　本研究の一部は JSPS 科研費 24700055 の助成を受け行われた.

参考文献

[1] 白子　準, 吉田宗弘, 押山直人, 和田康孝, 中野啓史, 鹿野裕明, 木村啓二, 笠原博徳：マルチコアプロセッサにおけるコンパイラ制御低消費電力化手法, 情報処理学会論文誌コンピューティングシステム, Vol. 47, No. SIG12(ACS15), pp. 147–158 (2006).
[2] 林　明宏, 和田康孝, 渡辺岳志, 関口　威, 間瀬正啓, 白子　準, 木村啓二, 笠原博徳：ヘテロジニアスマルチコア向けソフトウェア開発フレームワークおよび API, 情報処理学会論文誌コンピューティングシステム, Vol. 5, No. 1, pp. 68–79 (2012).
[3] Mei, J. and Li, K.: Energy-Aware Scheduling Algorithm with Duplication on Heterogenous Computing Systems, *Proceedings of ACM/IEEE 13th International Conference on Grid Computing*, pp. 122–129 (2012).
[4] Wang, L., Tao, J., von Laszewski, G. and Chen, D.: Power Aware Scheduling for Parallel Tasks via Task Clustering, *Proceedings of 16th International Conference on Parallel and Distributed Systems*, pp. 629–634 (2010).
[5] Wang, L., von Laszewski, G., Dayal, J. and Wang, F.: Towards Energy Aware Scheduling for Precedence Constrained Parallel Tasks in a Cluster with DVFS, *Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 368–377 (2010).
[6] Ito, M., Hattori, T., Yoshida, Y., Hayase, K., Hayashi, T., Nishii, O., Yasu, Y., Hasegawa, A., Takada, M., Ito, M., Mizuno, H., Uchiyama, K., Odaka, T., Shirako, J., Mase, M., Kimura, K. and Kasahara, H.: An 8640 MIPS SoC with Independent Power-Off Control of 8 CPUs and 8 RAMs by An Automatic Parallelizing Compiler, *2008 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 90–598 (2008).
[7] Howard, J., Dighe, S., Hoskote, Y., Vangal, S., Finan, D., Ruhl, G., Jenkins, D., Wilson, H., Borkar, N., Schrom, G., Pailet, F., Jain, S., Jacob, T., Yada, S., Marella, S., Salihundam, P., Erraguntla, V., Konow, M., Riepen, M., Droege, G., Lindemann, J., Gries, M., Apel, T., Henriss, K., Lund-Larsen, T., Steibl, S., Borkar, S., De, V., Wijngaart, R. V. D. and Mattson, T.: A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS, *2010 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 108–109 (2010).
[8] : Standard Task Graph Set Version 2, `http://www.kasahara.cs.waseda.ac.jp/schedule/`.
[9] Kasahara, H. and Narita, S.: Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing, *IEEE Transactions on Computers*, Vol. C-33, No. 11, pp. 1023–1029 (1984).