Vol.2013-ARC-207 No.14
Vol.2013-HPC-142 No.14
2013/12/16

使用コア数最適化とDVFSを用いたGPUの省電力化手法の検討

藤原 祐太[1]　松本 洋平[1]　和田 康孝[1]　近藤 正章[1]　本多 弘樹[1]

概要：近年，GPU を汎用的な計算に応用することが広く普及しているが，その消費電力の増加が問題となっている．本稿では，GPU のスイッチング電力削減手法として周波数と電源電圧を動的に変更する DVFS を利用し，またアイドルコアのリーク電力の削減に向け使用コア数の最適化を同時に行うべく，その際の性能，および消費エネルギーの傾向を解析する．特に，サイクルレベルのシミュレーションを用い，アプリケーションの挙動と消費エネルギーの内訳を含めた詳細な解析を行うことで，プログラムの特徴に応じた DVFS および使用コア数最適化手法の構築を目指す．解析の結果，周波数・電圧制御の効果はメモリアクセスや演算器部の負荷に依存し，コア数制御の効果は起動スレッド数や実際のスレッド並列度に依存することがわかった．

1. はじめに

近年，半導体の微細化技術の恩恵を受け，GPU の性能向上が著しい．従来，GPU はグラフィック処理を行うデバイスとして普及してきたが，その高い演算能力を HPC 分野などの汎用的な計算処理に応用することが一般的になってきている．

GPU は CPU に比べ大量の演算器を搭載しており，並列性の高い処理で高性能を得ることができる反面，演算回路の大規模化により消費電力の増加が問題となってきている．例えば，NVIDIA 社の GeForce GTX480 の消費電力は最大で 250W であり [1]，高性能な CPU と比較しても消費電力は同程度かそれ以上である．高性能計算環境においても，消費電力の増大は発熱量の増加による冷却コスト増加や信頼性の低下に繋がるため，低消費電力化は重要である．一方，近年ではモバイル端末においても，高度なグラフィック処理の要求の高まりから，比較的高性能な GPU が搭載されるようになってきている．例えば NVIDIA 社の Tegra4i は，4+1 コアの CPU と 60 コアの GPU が搭載されており [2]，大量の演算器を持つ GPU を備えたモバイル端末が普及しつつある．バッテリ駆動のモバイル端末においては GPU の消費エネルギー削減も重要な課題である．

一般的に，LSI の消費電力の内訳として，動的消費電力であるスイッチング電力と，静的消費電力であるリーク電力がある．スイッチング電力の削減技術として，LSI に要求される負荷に応じて電源電圧と動作周波数を変更する Dynamic Voltage and Frequency Scaling (DVFS) が広く使われており，動作周波数低下に比例して，また電源電圧低下の 2 乗に比例して電力を削減できるため，有効な省電力手法である．一方で，リーク電力削減技術としては，使用されていない回路領域への電源供給を遮断する Power Gating (PG) 技術が広く使われている．

GPU でも既に DVFS は実装されており，例えば GTX480 では，700MHz，400MHz，200MHz の 3 種類の動作周波数を設定可能である．また，GPU において DVFS を適用する場合の性能・電力モデル [4] や，メモリストール率に基づく周波数制御手法 [3] など，スイッチング電力削減に関する研究もいくつか行われている．ただし，アプリケーションの特徴と細粒度に周波数・電圧を変更させた場合の性能・エネルギー消費の傾向については十分に調査されていない．また，リーク消費電力については，コアがアイドル時にコア単位で PG を行う手法が存在する [7]．ただし，カーネル内のブロック数が GPU のコア数以上のものを実行する場合は，通常全コアが動作状態になり，スレッド数の不足やメモリアクセス遅延により演算器利用率が高くなくても，常に全コアでリーク電力が消費されてしまう．そのため，より積極的にコアを PG できる可能性もあると考えられる．

本稿では，GPU の性能と電力を評価できるシミュレータを用いて，細粒度に周波数と電圧を制御した場合，および使用コア数をスケーリングさせた場合の性能と消費エネ

[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.

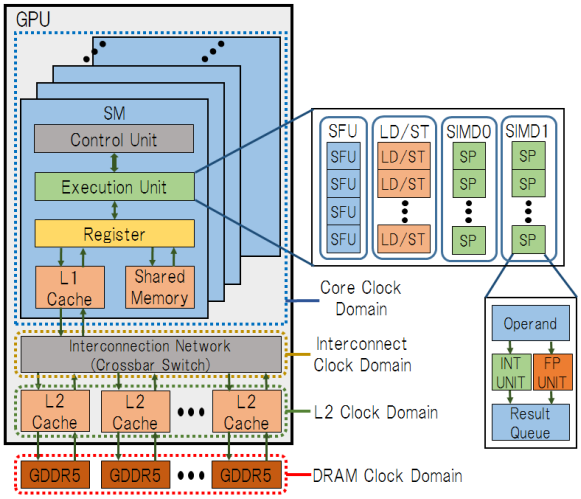


図 1 GPU のハードウェア構成

ルギーを測定し，アプリケーションの特徴とあわせてその傾向について議論する．そして，特に消費エネルギー削減要求の高いモバイル用途の計算機向けの DVFS とコア数スケーリング手法の構築に向け，プログラムの特徴に応じた周波数・使用コア数制御手法について検討する．

2. GPU における電力削減手法

本章では，GPU における DVFS と使用コア数変更による低消費エネルギー化の背景として，GPU のハードウェア構成と電力消費の傾向について述べる．

2.1 GPU のアーキテクチャモデル

図 1 は一般的な GPU のハードウェア構成を示している．GPU チップはいくつかの Streaming Multiprocessor (SM) と呼ばれるプロセッサコアと，コアとメモリを繋ぐ Interconnection Network，L2 Cache で構成されている．SM は内部に Control Unit，Execution Unit，Register，L1 Cache，Shared Memory を持つ．Execution Unit は，整数および浮動小数点演算を行う SIMD Unit，特殊な演算を行う SFU，ロードストアを実行する LD/ST ユニットから構成されており，SIMD Unit の 1 つには 16 個の演算器である Streaming Processor (SP) があり，命令に応じて整数ユニットと浮動小数点ユニットが使い分けられる．また，オフチップの主記憶として GDDR5 DRAM が GPU チップに接続される．

GPU には SM 部, Interconnection Network 部, L2 Cache 部，DRAM 部の複数のクロックドメインが存在し，それぞれ異なるクロック周波数で動作する．本稿で検討する DVFS は，SM 部 (図 1 ”Core Clock Domain”部) のみ周波数と電源電圧を変更することを想定し，その他のドメインは周波数・電圧とも一定として評価を行う．

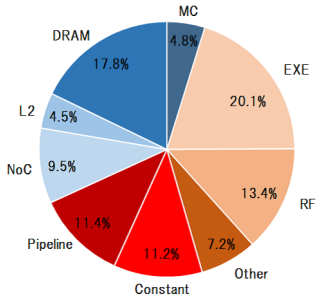


図 2 GPU の消費電力の内訳 (出展 [4])

表 1 消費電力の内訳中の各コンポーネント

Func	演算器使用による電力
RF	レジスタファイルアクセスの電力
Others	GPU チップのその他機構の電力
Const	プログラム実行に関係なく消費される静的電力
Pipe	パイプラインの電力
NoC	Interconnection Network の電力
L2	L2 キャッシュアクセスの電力
DRAM	DRAM アクセスの電力
MC	メモリコントローラの電力

2.2 GPU の電力消費傾向

図 2 は文献 [3] で報告されている GPU の各コンポーネント (表 1 参照) の平均消費電力の割合を示している．本データは，3.1 節で述べるシミュレーション環境を用い，プログラムを動作させた際の平均消費電力を複数のベンチマークで平均化したものである．

図 2 より GPU の消費電力の約 52%がコア部のスイッチング消費電力であることがわかる．そのため，まずはコアの消費電力削減を考えることが，GPU の消費エネルギー削減の効果が大きいことがわかる．一方で，主にリーク電力に起因すると考えられるプログラム実行に関係なく消費される電力 (Const) も約 11%と無視できない．特に，アプリケーションによっては GPU に多数のコアが存在していても，それら全てを使い切れないものも多く，アプリケーションの特性に応じて使用コア数を変更し，未使用コアを PG することで消費エネルギーを削減できる可能性がある．

2.3 関連研究

近年，GPU における周波数・電圧制御に関する研究が行われるようになってきた．文献 [5] では，GPU コアの周波数とメモリ周波数をそれぞれ Min-Low-High の 3 段階に分け，それらを適切に設定することで GPU の省電力化を狙う手法を提案している．リアルシステムにおいて，1%未満の性能低下率で 28%のエネルギーが削減できると報告されている．また，NVIDIA GeForce GTX465(11 コア) と NVIDIA GeForce GTX480(15 コア) の実チップを用い，周波数を 50MHz 毎に静的に変更した際の平均消費電力，実行時間，温度の測定結果も報告されている [1]．Xinxin ら [6]

表 2　仮定した GPU のパラメータ

コア数	15
Thread 数/コア	1536
SIMD Unit 数/コア	2
SIMD Pipeline 数	32
Register size/コア	128[KB]
Shared Memory size/コア	16[KB]
L1 Cache size/コア	48[KB]
L2 Cache size/コア	768[KB]
Core Clock	700[MHz]
Interconnect Clock	1400[MHz]
L2 Clock	700[MHz]
DRAM Clock	924[MHz]

は，GPU において DVFS により，性能低下を 4%以下と設定した場合，平均で 19.28 %のエネルギー削減に成功している．ただし，コアとメモリの周波数制御の効果は，アプリケーションの特性に依存すると報告されている．

パフォーマンスのプロファイルを用いて GPU の消費電力をモデリングする研究もなされている [4][9]．文献 [4] は周波数を 3 段階，電圧を 2 段階に分け，消費電力の傾向を測定しつつ電力の推定を行っている．また，文献 [9] では，電力モデルでの電力の平均誤差率は 4.7%であり，高精度な推定を実現できている．さらに，パフォーマンスカウンターを用いた動的周波数制御など電力最適化手法への応用が期待できると述べられている．

GPU における PG 手法も検討されており，Wang ら [7] は GPU のコア単位で PG の可能性を調査している．PG を用いることで，約 60 %のリーク電力の削減に成功している．

3. GPU の性能と電力評価環境

本稿では，サイクルレベルのシミュレータにより DVFS と実行時に使用コアをスケーリングさせた場合の性能と電力を評価する．本章ではその評価環境と，評価に用いた仮定について述べる．

3.1 評価環境

本稿では，GPU の性能評価に GPU のサイクルレベルシミュレータである GPGPU-Sim(version 3.2.0)[8] を用いて評価を行う．GPGPU は CUDA や OpenCL で記述されたアプリケーションを評価でき，コア数やレジスタサイズなどのハードウェアパラメータを変更して性能を評価できる．今回の評価では，ベースとする GPU のハードウェアモデルとして NVIDIA GeForce GTX480 を仮定した．表 2 に設定したパラメータを示す．表中の Core Clock はベースとなる GPU の最高周波数を意味している．

消費電力は，GPGPU-Sim に付随する GPUWattch[3] を用いて評価する．GPUWattch は GPGPU-Sim で得られる各構成要素に対する利用回数やアクセス回数といったカウンタの値をもとに，McPat[11] の電力モデルを利用して一定サイクル毎に各構成要素の消費電力を求めるものである．最終的には，アプリケーション実行の平均消費電力を算出し，それに実行時間を掛けることで消費エネルギーを求める．なお，温度に関しては 380 ケルビンに設定して評価する．

表 3　動作周波数・電源電圧の組合せの仮定

周波数 [MHz]	電圧 [V]
700	1.000
600	0.925
500	0.850
400	0.775
300	0.700
200	0.625
100	0.550

使用するベンチマークについては，ISPASS2009 ベンチマーク [8] より CUDA で記述された 6 つのアプリケーションプログラムを用いて評価を行う．なお，評価結果には CPU と GPU 間のデータ転送時間や，その際の消費電力は含めないものとする．

3.2 電力評価モデル

GTX480 のデフォルトのコアクロックは 700MHz で，その際の電源電圧は 1.0V であり，また最低周波数である 100MHz 動作時の電源電圧は 0.55V である [3]．本稿では，多くの周波数・電源電圧が設定できると仮定して評価を行うため，100MHz 刻みで周波数が設定できると仮定し，その際の電源電圧は 1.0V と 0.55V を線形補完した値を仮定する．設定可能な周波数・電源電圧の組み合わせを表 3 に示す．

なお，先に述べたように本稿の評価では，図 1 の Core Clock Domain 部分のみの動作周波数と電源電圧を変更する．当該ドメインは周波数と電源電圧の 2 乗に比例して消費電力が変化するが，他のドメインの電力はそれらには依存しない．ただし，コアの周波数に応じて単位時間あたりのアクセス回数が変化し得るため，消費電力は Core Clock Domain の周波数に応じて変化する可能性もある．また，メモリアクセス待ちなどによりコアがアイドル時にはリーク電力が消費されるが，その電力は電源電圧の 1 乗に比例すると仮定して評価を行う．

使用コア数に関しては，もともとある 15 コアを 1 コアずつ停止させることを想定して評価を行った．コア数はアプリケーション実行中には変化させず，最低のコア数は 2 とした．この際に，使用していないコアに関しては完全に Power-Gating ができるものとした (リーク電力は 0 になる)．なお，停止されていないコアが実行待ちによりアイドルになった際には，リーク電力のみが消費される．

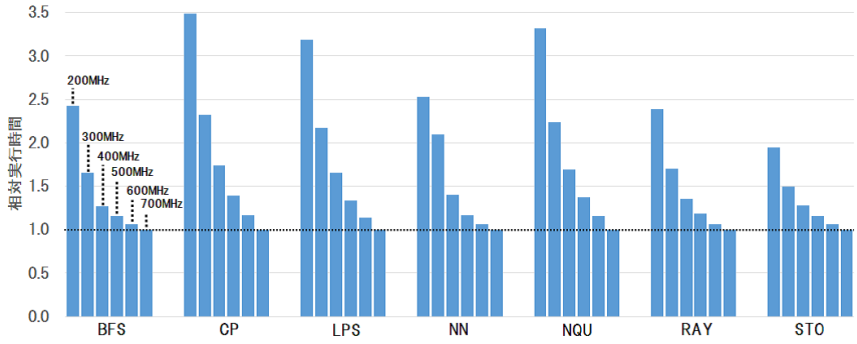


図 3 アプリケーション毎の相対実行時間

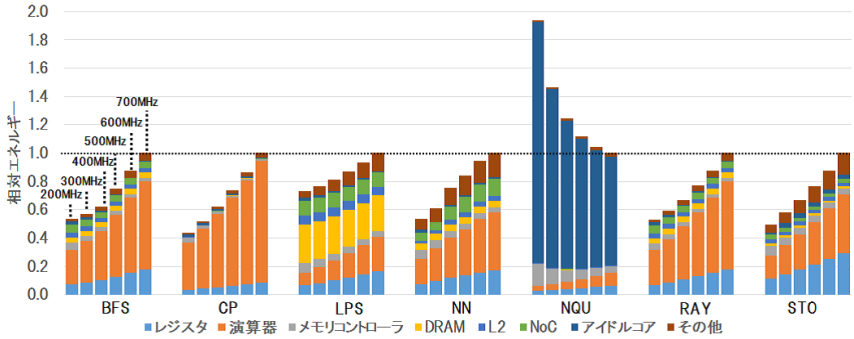


図 4 アプリケーション毎の消費エネルギーの内訳

4. 周波数とコア数変更による GPU の電力削減効果の評価

ここでは，周波数・電源電圧，並びに使用コア数を制御した際の性能と消費エネルギーに与える影響を調査するため，静的にそれらを制御して評価を行い，アプリケーションの特徴とあわせてそれらの傾向について議論する．

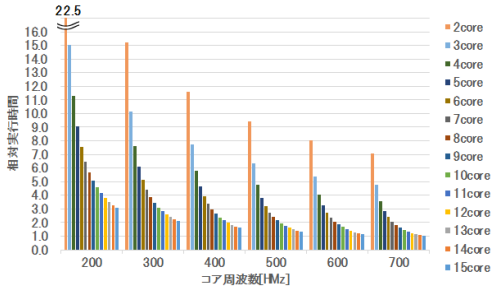
4.1 周波数制御の評価結果

図 3 は使用コア数を 15 個とし，Core Clock Domain について動作周波数を 700MHz から 200MHz まで 100MHz 刻みで低下させた場合の実行時間を示している．なお，実行時間は各アプリケーション毎に 700MHz を基準とした相対実行時間である．各アプリケーション毎に 6 つの棒グラフがあり，左から動作周波数が 200MHz から 700MHz までの結果を示している．図 4 は，図 3 の各場合の消費エネルギーの評価結果である．同様に各アプリケーションで動作周波数を 700MHz として実行した際の消費エネルギーに対する相対消費エネルギーを示している．また，各棒グラフは消費エネルギーの内訳を示しており，下からレジスタファイルアクセス，演算器使用，メモリコントローラ，DRAM アクセス，Interconnection Network，アイドルコア，その他機構の消費エネルギーである．

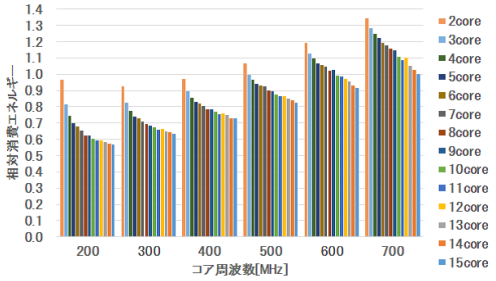
図 3 より，周波数を下げることによって全てのアプリケーションの実行時間が増加している．基本的にクロック周波数は性能に比例するため当然の結果であり，CP，NN，NQU などは 700MHz の場合に対して周波数にほぼ比例して性能が低下している．しかし，その他のアプリケーションでは周波数低下分に比べて性能低下率は大きくない．プロセッサにおける DVFS 手法でも多くの報告があるように [8]bib10，メモリアクセスなどがボトルネックとなる場合には，演算ドメインの周波数を下げても性能低下が大きくないことと同様の傾向である．

消費エネルギーに着目すると，NQU を除く全てのアプリケーションで周波数を下げることにより消費エネルギーが削減されている．これは，スイッチング消費電力は周波数の 1 乗に，また電源電圧の 2 乗に比例することから，実行時間が周波数分長くなったとしても消費エネルギー削減の効果があるためである．また，BFS や CP，RAY，STO はの消費エネルギー低下率は他のアプリケーションに比べても高い．これは，演算器やレジスタファイルなど，周波数・電源電圧の制御対象ドメインにある機構の消費電力が大きく，DRAM や NoC など制御対象ドメイン以外の消費電力が少ないため，周波数・電圧制御の恩恵を大いに受けることができているためである．

一方で，NQU では周波数を低下させることで消費エネルギーが増加してしまっている．NQU はアイドル状態のコア数が多く，一部のコアのみが動作するという特徴がある．そのため，周波数・電源電圧の制御対象ドメインのスイッチング消費電力が小さく，周波数制御の効果が少ないため，周波数低下で実行時間が長くなると合計の消費エネルギーが増加してしまうことが理由であると考えられる．

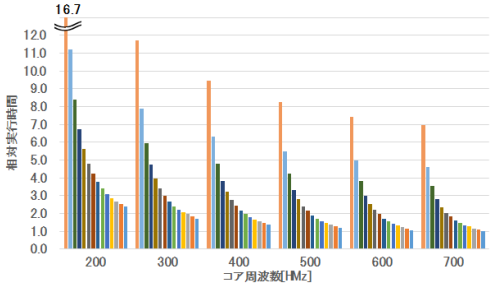


(a) NN の相対実行時間

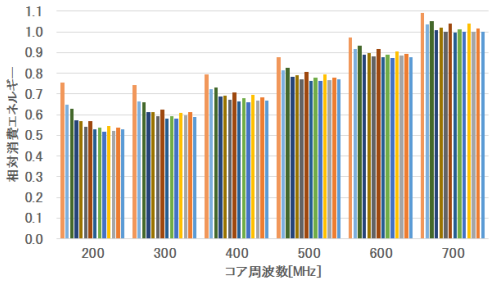


(b) NN の消費エネルギー

図 5 NN の評価結果

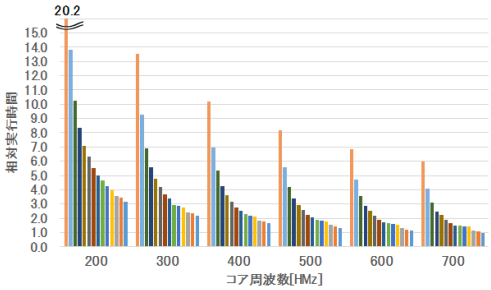


(a) RAY の相対実行時間

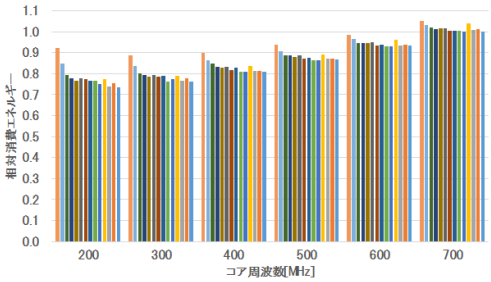


(b) RAY の消費エネルギー

図 6 RAY の評価結果

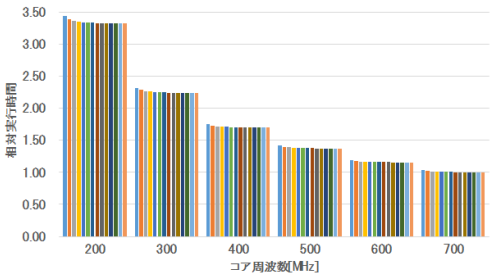


(a) LPS の相対実行時間

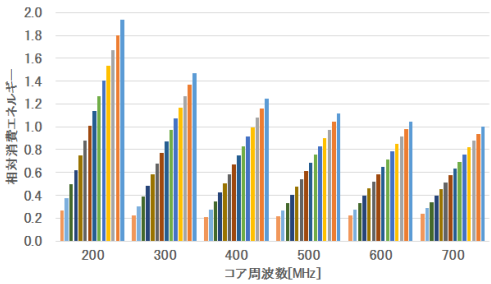


(b) LPS の消費エネルギー

図 7 LPS の評価結果



(a) NQU の相対実行時間



(b) NQU の消費エネルギー

図 8 NQU の評価結果

図 4 でも，NQU のアイドルコアによる消費エネルギーが周波数低下に従い増加していることが示されている．

以上の結果をまとめると，多くのアプリケーションにおいては周波数を制御することが消費エネルギー削減には有効であることが確認できた．ただし，周波数低下に応じて性能が低下してしまうアプリケーションもあり，性能が最も重要となる HPC 分野での利用は制限されてしまう可能性がある．しかし，モバイル端末など消費エネルギーの削減が重要な分野では，GPU における DVFS 手法は有効な手法になり得ると考えられる．

4.2 使用コア数制御の評価結果

本節では，使用コア数を制御した場合の性能と消費エネルギーの傾向に関して評価を行う．実験に用いたアプリケーションの中から，異なる傾向を示すものを代表して，NN，RAY，NQU，LPS の評価結果を用いて議論を行う．

図 5，図 6，図 7，および図 8 は，それぞれ使用コア数と周波数を変更させつつ NN，RAY，LPS，NQU の各プログラムを実行した際の実行時間と消費エネルギーの評価結果である．実行時間と消費エネルギーは使用コア数が 15 個，動作周波数が 700MHz の場合に対する相対値を示してい

る．各周波数毎に 14 個の棒グラフがあり，最右がコア数 15 個の場合で左に向かうにつれ使用コア数を 1 つづつ減らして実行した場合の相対実行時間と相対消費エネルギーを示している．

まず実行時間について議論する．NN，RAY，LPS は使用コア数を削減すると実行時間が長くなる．これらは並列性の高いアプリケーションであり，多くのコアを利用して並列処理を行うことで高性能化が達成できるアプリケーションであることがわかる．また，前節の 15 コアの場合の周波数制御の評価結果と同様に，それぞれのコア数同士を比較した場合に周波数を低下させると実行時間が長くなっている．ただし，LPS は NN，RAY に比べるとコア数を削減した際の性能への影響が小さい．これは起動スレッド数が少ないためである．一方で NQU は一部のコアのみが動作するという特徴から，コア数を削減しても実行時間にほとんど影響がない．

次に消費エネルギーについて議論する．NN は特に並列性の高いアプリケーションであり，コア数の削減に比例して実行時間が長くなるため，コア部以外の電力による消費エネルギーが増加してしまい，コア数を減らすと消費エネルギーが大きく増加してしまう．このようなアプリケーションでは GPU に搭載されたコアを最大限利用し，必要に応じて周波数・電源電圧制御を行うことが消費エネルギーの観点からは有効である．

RAY と LPS では，コア数を削減しても消費エネルギーはほとんど変化しないか，周波数に依存してコア数が 10 コア前後の際に僅かであはあるが消費エネルギーが最小値をとる場合がある．ただし，RAY と LPS では周波数を制御した際の消費エネルギーへの影響が異なり，RAY の方が周波数・電源電圧を削減したときの消費エネルギー削減効果が大きい．RAY は演算による電力が大部分を占めることから，コア数を削減した分だけ電力が削減されるが，その分実行時間が増加することで，コア数の違いによる消費エネルギーの影響がほとんどないと考えられる．一方で，LPS は演算部以外の電力も大きく，コア数削減により実行時間が長くなるとエネルギーが増加する可能性があるが，コア数削減分に対して性能低下率が小さいため，それらの影響が相殺されて，コア数の違いによる消費エネルギーの影響が見えなかったと考えられる．これらのようなプログラムでは，最適なコア数を選択することで，多少ではあるが消費エネルギーを削減できる可能性があると考えられる．

NQU は，コア数を削減することで消費エネルギーが大幅に削減されている．NQU はコア数を減らしても実行時間が変わらないこと，また，消費エネルギーの大部分をアイドルコアによるものが占めていることから，コア数削減によりアイドル電力を大幅に削減できたことが消費エネルギー削減に寄与している．このように並列性の低いプログラムでは，コア数制御が消費エネルギーに有効である．

表 4 アプリケーション毎の特性と制御の関係

	アプリケーション特性				制御	
	演算器	メモリアクセス	スレッド	アイドルコア	周波数・電圧	コア数
BFS	high	middle	high	low	○	△
CP	middle	low	middle	low	○	×
LPS	high	high	low	low	△	△
NN	low	middle	low	low	○	×
NQU	low	low	low	high	×	○
RAY	middle	high	high	low	○	△
STO	high	low	middle	low	○	×

4.3 考察

前節までの評価から，アプリケーションの特徴，および周波数・電源電圧制御とコア数制御に関しての消費エネルギー削減への有効性を表 4 に示す．

表 4 は，アプリケーションの特徴を演算器使用率，メモリアクセス頻度，スレッド数，並びにアイドルコア数について，多い順にそれぞれ high，middle，low で示している．また，周波数・電源電圧制御とコア数制御の有効性を○，△，×の記号で示している．

評価の結果から，多くのアプリケーションにおいて周波数・電源電圧制御の有効性が期待できるが，一方でアイドル状態のコア数が多いアプリケーションにおいては，周波数・電源電圧を下げることで消費エネルギーが増加したことから，周波数を低下させる上での判断材料の 1 つとして，アイドルコア数を用いることが考えられる．また，性能低下が大きいアプリケーションも多いことから，性能低下の閾値を設けるなどの制御が必要になると考えられる．

コア数のスケーリングに関しては，並列性の高いアプリケーションで使用コア数を削減してしまうと実行時間の増大を招くだけでなく，場合によっては消費エネルギーが増大してしまう場合もあり注意が必要である．一方で，アイドルコア数が多いアプリケーションでは，消費エネルギーを削減できる可能性があるため，こちらもアイドルコア数が一つの指標になり得ると考えられる．

5. まとめ

本稿では，GPU において DVFS と使用コア数の最適化手法の構築を目指し，それらが消費エネルギーに与える影響とアプリケーションの特徴との関係を詳細に解析することを目的に，サイクルレベルシミュレーションを用いて性能，および消費エネルギーの評価を行った

評価の結果，多くのアプリケーションにおいて，周波数・電源電圧制御は消費エネルギー削減に有効であることがわかった．特に，メモリアクセスが少ないアプリケーションにおいては大幅な消費エネルギー削減を確認できた．ただし，性能低下が大きいアプリケーションも多いことがわかった．使用コア数の制御において最も効果が大きいのは，アイドルコアが多いアプリケーションであった．また，スレッド数の少ないアプリケーションに関しては多少のエネ

ルギー削減効果があることがわかった．

今後の方針として，アプリケーションの特性に応じて動的に周波数と使用コア数の制御を行う手法を開発する予定である．また，DRAM や NoC，キャッシュなども含めた消費エネルギー削減手法を検討することも今後の課題である．

謝辞 本研究の一部は，科学技術振興機構・戦略的創造研究推進事業 (CREST) の研究プロジェクト「ポストペタスケールシステムのための電力マネージメントフレームワークの開発」，ならびに JSPS 科研費 24700055 の助成により行われたものである．

参考文献

[1] J. M. Cebrian, G. D. Guerrero, and J. M. Garcia. Energy Efficiency Analysis of GPUs, Proc. PDPSW'12 , pp.1014-1022 , 2012.
[2] NVIDIA. http://blogs.nvidia.com/blog/2013/02/25/how-phoenix-the-tegra-4i-reference-phone-will-bring-awesome-features-to-the-mainstream/
[3] J. Leng et al. GPUWattch: Enabling Energy Optimizations in GPGPUs. Proc. ISCA'13, pp.23-27, 2013.
[4] 長坂 仁，丸山 直也，額田 彰，遠藤 敏夫，松岡 聡．" GPU におけるモデルに基づいた電力効率の最適化"，情報処理学会研究報告．2010-HPC-128, pp.1-6 , 2010-12
[5] Y. Abe et al. Power and performance analysis of GPU-accelerated systems. Proc. HotPower'12, pp.1-5, 2012.
[6] X. Mei et al. A measurement study of GPU DVFS on energy conservation. Proc. HotPower'13, Article No. 10, 2013.
[7] P. Wang et al. Power gating strategies on GPUs. ACM Transactions on Architecture and Code Optimization (TACO) TACO Homepage archive Volume 8 Issue 3 Article No. 13 , 2011
[8] A. Bakhoda et al. Analyzing CUDA workloads using a detailed GPU simulator. Proc. ISPASS-2009, pp.163-174 , 2009.
[9] H. Nagasaka et al. Statistical Power Modeling of GPU Kernels Using Performance Counters. Proc. IGCC'10, pp.115-122, 2010.
[10] S. Hong, and H. Kim. An integrated GPU power and performance model. Proc. ISCA'10, pp.280-289, 2010.
[11] S. Li et al . McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. Proc. MICRO-42, pp469-480, 2009.