Vol.2010-ARC-192 No.2
Vol.2010-HPC-128 No.2
2010/12/16

ＧＰＵにおけるモデルに基づいた電力効率の最適化

長坂　仁[†1]　　丸山直也[†1]　　額田　彰[†1]
遠藤敏夫[†1]　　松岡　聡[†1,†2]

ＧＰＵの演算性能の飛躍的な発達により、グラフィックス処理だけでなく汎用計算に用いられるようになるにつれて、ＧＰＵの消費電力削減の重要性が増している。そこで我々はまずＧＰＵの消費電力予測の必要性とその結果について述べ、さらに、消費エネルギー削減に向けた第一歩として電圧と動作周波数を変更することで消費電力がどのように表現できるかを解析した。その結果、ＧＰＵにおける消費電力は定常状態における静的な要素 (P_{static}) と、プログラム実行による増加分の動的な要素 ($P_{dynamic}$) とに分けてることができると仮定した場合、P_{static} は電圧値に、$P_{dynamic}$ は電圧値の２乗と動作周波数にそれぞれ比例して変化している結果を得た。これにより、先の消費電力予測と合わせる事で様々な環境でのＧＰＵにおける消費電力を高精度で予測することが可能となった。

Optimization of electric power efficiecy based on model in GPU

Hitoshi Nagasaka,[†1] Naoya Maruyama,[†1]
Akira Nukada,[†1] Toshio Endo[†1]
and Satoshi Matsuoka[†1,†2]

GPUs are being employed in large-scale supercomputing environments,where their power consumption is a first-class design constraint.to reduce their power consumption, we present a prediction model that leverages appliction behavior behavior observable through performance counters. In addigion, how power consumption is expressible by the change of the Voltage and the operating frequency as the first step for the consumption energy reduction is analyzed. When the result power consumption in GPU is able to be devided into static element(P_{static}) in the stationary state and element($P_{dynamic}$) dynamic for an increase by the program execution, $P_{dynamic}$ is proportional to f and square of V and P_{static} is proportional to V. With our model, we show that GPU power in varying frequency and voltage conditions can be accurately estimeted.

1. はじめに

近年、ＧＰＵの高い演算処理能力を従来のグラフィックス処理以外の汎用計算に応用する技術（ＧＰＧＰＵ）が注目されている。その応用範囲は多岐に渡り、具体的には波の動きなどの大規模な流体計算[1)] や生物情報学[2)]、や医療分野[3)] などが挙げられる。又、本学のスーパーコンピュータＴＳＵＢＡＭＥにも搭載されており、今後もＧＰＵのＨＰＣ分野での利用は広がっていくと考えられる。そのＧＰＵの利用の拡大につれ、ＧＰＵの低消費電力化は非常に重要な問題になっている。[4)] 具体的には、今回の実験で用いたＧＰＵである NVIDIA GeForce GTX480 では１枚のＧＰＵの消費電力は最大で 250W 程度にもなり、一般的なＣＰＵの消費電力を優に超えている。我々はまずＧＰＵの低消費電力化の為、その消費電力の特徴を調査した。カーネル関数を実行して得られるパフォーマンスカウンタの値から消費電力を線形回帰分析により予測した。さらに、消費エネルギー削減への第一歩としてＣＰＵの消費エネルギー削減手法には広く用いられている「DVFS」に注目し、電圧値と動作周波数を手動で変更して消費電力の振る舞いを解析した。その際に消費電力を定常状態を表す部分とプログラム実行による増加分を表す部分に分けそれぞれが、電圧値に比例、電圧値の２乗と動作周波数に比例していることを示す結果を得た。これらの事から、様々な実行プログラムにおいて、設定した電圧値と動作周波数でのＧＰＵにおける消費電力を正確に予測することが可能となった。

2. ＧＰＵにおける消費電力

2.1 ＧＰＵにおける電力の取得法

我々は電流計には株式会社シナジェティック製ＳＴ-30600 を使用する。計測の際に配線を加工が不要なクランプセンサを用いている。

ＧＰＵには主に「ＡＴＸ電源の 12 Ｖ線から供給される電力」と「ＰＣＩＥｘｐｒｅｓｓからの供給される電力」の２箇所から取得しており、両者の電力を測定する必要がある。前者の測定には 12 Ｖ線に電流センサを装着することで測定が可能となる。一方、ＰＣＩｅか

†1 東京工業大学
Tokyo Institute of Technology
†2 国立情報学研究所
National Institute of Informatics(NII)

Vol.2010-ARC-192 No.2
Vol.2010-HPC-128 No.2
2010/12/16

表 1 回帰に用いたカウンタ

Name	Descriptio
gld_32	32-byte global memory load transactions
gld_64	64-byte global memory load transactions
gld_128	128-byte global memory load transactions
gst_32	32-byte global memory store transactions
gst_64	64-byte global memory store transactions
gst_128	128-byte global memory store transactions
local_load	Local memory load
local_store	Local memory store
branch	Number of branches
divergent_branch	Number of divergent branches
instructions	Instructions executed
shared_load	Number of executed shared load instructions
shared_store	Number of executed shared store instructions
l1_global_load_miss	Number of global load hits
l1_global_load_hit	Nubber of global load misses

ら供給される電力はライザーカードを挟み、その中から 12 V、3.3 Vの電力を供給している配線を測定した。又、計測時のサンプリング間隔は１ｍｓとした。

2.2 消費電力予測

我々はＧＰＵ上でのカーネル関数における消費電力とそのカーネル実行のパフォーマンスカウンタ[6] との相関を線形回帰分析にかけて予測した。尚、パフォーマンスカウンタは CUDA Profiler を用いて取得し、今回回帰分析に用いたカウンタを 1 に示す。

また、その精度を解析するために 10fold-Cross Validation を用いた。実験に使用したコードは CUDA SDK 付属のサンプルコード、rodinia ベンチマーク[7]、FFT プログラムである。また、元のコードではカーネル関数が非常に短いものが多い為、計測誤差を小さくする為にカーネル内の処理を繰り返し行い、全てのカーネルの実行時間が１秒間となるように変更を施した。実測値と予測値を比較したグラフを図 1 に示す。正確に予測ができていることがわかる。

3. ＧＰＵにおける消費電力の振る舞い

3.1 アプリケーション実行時における消費電力の変化

以下に、CUDA SDK 付属のサンプルコードに含まれる nbody アプリケーションを実行した際の電力変化を示す。

表 2 を見て分かる通り、定常状態からプログラムを実行し電力が最大となった後、すぐに

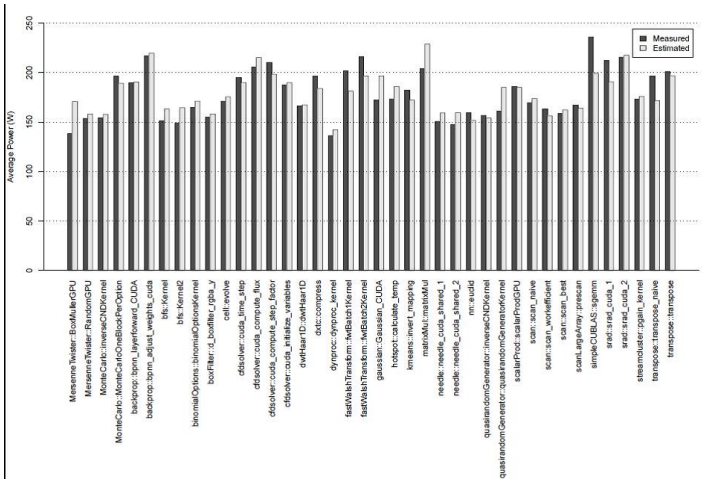

図 1 予測結果と計測値

定常状態には戻らない。我々はこの電力に注目し、解析を行う。

3.2 消費電力のうちわけ

前節の表 2 よりＧＰＵにおける消費電力は、等式 (1) のように２つに分けて考えることができる。

$$P = P_{dynamic} + P_{static} \quad (1)$$

以後本論分では様々な実験とその観測結果から図中のＢの電力を定常状態の電力 (P_{static})、図中のＡの電力をプログラム実行による増加分の電力 ($P_{dynamic}$) と仮定して議論する。

3.3 消費電力の振る舞い

我々はＧＰＵの消費電力削減の手法として、ＣＰＵで広く用いられている「ＤＶＦＳ」に着目しＧＰＵにおける電圧値と動作周波数を変更することで消費電力と性能がどのような振る舞いをするかを調査した。

4. 準備・実験

4.1 実 験 環 境

実験環境は表 2 の通り。

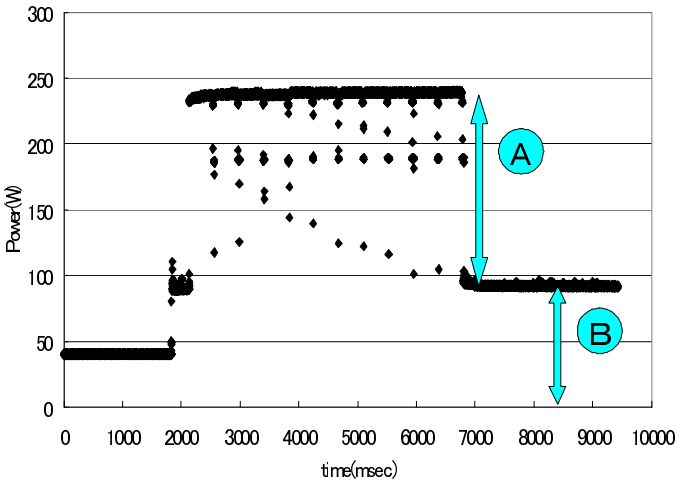

図 2 nbody における電力の振る舞い

表 2 実験環境

GPU	NVIDIA GeForce GTX 480
CPU	AMD Phenom 9850 Quad-Core Processor 2.5GHz
Memory	4GB
OS	WINDOWS 7(64bit 版)
GPUDriver	260.99
Compiler	gcc4.4.4 and CUDA Toolkit 3.1

4.2 計　　測

実験に使用したコードは大きな特徴のあるプログラムとして nbody(ほぼ計算のみを行う) と memcopy(ほぼメモリ転送のみを行う) を、一般のアプリケーションとして rodinia ベンチマークより bfs、ndyproc、srad を用いた。

また、電圧 (V) を default 値の 1037mV と 10%下げた 950mV(down) の場合、さらに、動作周波数 (f) を default 値 (coreClock:700MHz,memoryClock:1686MHz)、10%上げた (up) 値 (coreClock:770MHz,memoryClock:2033MHz)、10%下げた (down) 値 (coreClock:700MHz,memcClock:1666MHz) の 3 段階にてそれぞれの組み合わせで測定を行った。

表 3 実験結果

program	Voltage	f:down	f:default	f:up
nbody	default	P_d=129.01,P_s=89.92 t=0.471(sec)	P_d=144.73,P_s=91.71 t= 0.423(sec)	P_d=159.66,P_s=93.70 t=0.386(sec)
	down	P_d=104.68,P_s=80.81 t=0.471(sec)	P_d=117.28,P_s=82.47 t=0.423(sec)	P_d=129.61,P_s=84.04 t=0.386(sec)
memcopy	default	P_d=67.60,P_s=89.93 t=4.969(msec)	P_d=74.30,P_s=91.75 t=4.399(msec)	P_d=80.40,P_s=93.85 t=4.082(msec)
	down	P_d=58.14,P_s=80.95 t=4.901(msec)	P_d=63.66,P_s=82.65 t=4.367(msec)	P_d=69.23,P_s=84.15 t=4.073(msec)
bfs	default	P_d=49.46,P_s=88.77 t=43.38(msec)	P_d=54.78,P_s=90.57 t=40.92(msec)	P_d=59.99,P_s=92.34 t=37.34(msec)
	down	P_d=40.56,P_s=80.00 t=44.48(msec)	P_d=45.05,P_s=81.39 t= 41.01(msec)	P_d=49.17,P_s=83.11 t=37.67(msec)
dynproc	default	P_d=36.99,P_s=88.38 t=82.93(msec)	P_d=41.63,P_s=90.32 t=74.24(msec)	P_d=45.89,P_s=92.10 t=68.47(msec)
	down	P_d=30.12,P_s=79.56 t=82.42(msec)	P_d=33.78,P_s=81.09 t=74.76(msec)	P_d=37.34,P_s=82.65 t=68.03(msec)
srad	default	P_d=91.04,P_s=90.07 t=7.480(sec)	P_d=102.17,P_s=91.80 t=6.722(sec)	P_d=112.00,P_s=93.72 t=6.126(sec)
	down	P_d=75.44,P_s=80.85 t=7.480(sec)	P_d=84.43,P_s=82.55 t=6.723(sec)	P_d=92.64,P_s=83.97 t=6.126(sec)

表 4 性能比較 (演算性能 (GFLOPS) とバンド幅 (MB/sec)))

	Voltage	f:down	f:default	f:up
performance	default	640.385	712.302	781.795
	down	640.327	712.195	781.801
bandwidth	default	108.05	122.04	131.53
	down	109.54	122.94	131.8

4.3 測 定 結 果

それぞれの測定した平均消費電力と実行時間の結果を表 3 示す。
また、nbody と memcopy を実行した際に測定した時の性能とバンド幅を表 4 に示す。

5. 評　　価

5.1 $P_{dynamic}$ について

表??より $P_{dynamic}$ は実行プログラムによって値が大きく異なることが分かる。次に、電圧値と動作周波数を変更させてどのように変化したかを示す為に、それぞれの値を一定とし

Vol.2010-ARC-192 No.2
Vol.2010-HPC-128 No.2
2010/12/16

表 5 電圧値と動作周波数を変更した際の $P_{dynamic}$ の変化の比

		V を固定				f を固定		
program	Voltage	down	default	up		down	default	up
nbody	default	0.891	1	1.103		1	1	1
	down	0.893	1	1.105		0.811	0.810	0.812
memcopy	default	0.909	1	1.082		1	1	1
	down	0.913	1	1.088		0.860	0.857	0.861
bfs	default	0.903	1	1.094		1	1	1
	down	0.900	1	1.092		0.820	0.822	0.821
dynproc	default	0.889	1	1.102		1	1	1
	down	0.892	1	1.105		0.814	0.811	0.814
srad	default	0.891	1	1.096		1	1	1
	down	0.893	1	1.097		0.829	0.826	0.827

表 6 電圧値と動作周波数を変更した際の P_{static} の変化の比

		V を固定				f を固定		
program	Voltage	down	default	up		down	default	up
nbody	default	0.981	1	1.021		1	1	1
	down	0.980	1	1.019		0.899	0.899	0.897
memcopy	default	0.980	1	1.023		1	1	1
	down	0.979	1	1.018		0.900	0.901	0.897
bfs	default	0.980	1	1.020		1	1	1
	down	0.983	1	1.021		0.901	0.899	0.900
dynproc	default	0.979	1	1.020		1	1	1
	down	0.981	1	1.019		0.900	0.898	0.897
srad	default	0.981	1	1.021		1	1	1
	down	0.979	1	1.017		0.898	0.899	0.896

たときの消費電力を 1 とした比を表 5 に示す。横軸の down、default、up は動作周波数を表しその値は表 3 と同じである。

表**??**より、$P_{dynamic}$ は電圧を固定した時動作周波数を 10%変動させると消費電力も約 10%変動していることが分かる。これより $P_{dynamic}$ は動作周波数に比例していると推測される。又、動作周波数を固定した時に電圧を 10%変動させると消費電力は 20%近く変動していることが分かる。つまり、電圧の 2 乗に比例していると推測される。(∵ 1.10x1.10=1.21(約 20%))

これらのことから $P_{dynamic}$ は以下のように表すことができると推測できる。

$$P_{dynamic} = l * V^2 * f \quad (l : const.) \tag{2}$$

5.2 P_{static} について

同様に P_{static} についても考察する為、消費電力の比を表 6 に示す。

表**??**より、P_{static} は電圧を固定した時に動作周波数を 10%変動させても消費電力も数%のみしか変動していない。これより P_{static} は動作周波数に依存しない電力であると推測される。この数%の誤差に関しては後述する。又、動作周波数を固定した時に電圧を 10%変動させると消費電力も約 10%変動していることが分かる。これより P_{static} は電圧値に比例していると推測される。

これらのことから P_{static} は以下のように表すことができると推測できる。

$$P_{static} = k * V \quad (k : const.) \tag{3}$$

5.3 誤差について

動作周波数を変更した際に生じた数%の誤差であるが、以下のような原因が挙げられる。

(1) P_{static} には $P_{dynamic}$ の一部が含まれている可能性がある

(2) 発熱量が変わる場合、ファンの回転速度も変わりファンによる消費電力が変わる可能性がある

(3) 電圧値、動作周波数を変更した場合、チップの温度も変わるため消費電力も変わる可能性がある

(2) や (3) が原因とすると数ワットの誤差を正確に推定することは非常に困難である。(1) の場合、その含まれる電力について解析する必要がある。実際に、P_{static} として含まれていた電力の一部を $P_{dynamic}$ と仮定した処動作周波数を変更した場合の数値が正確に表現できた。この調査に関しては今後の課題とする。

5.4 実行時間と消費エネルギーについて

表 4 からも分かる通り、実行時間は電圧とは関係が見られず、動作周波数に比例していることが分かる。表 7 にそれぞれのプログラムの消費エネルギーを示す。

また、消費エネルギーに関しては以下のようなことが言える。ここでは電圧値、動作周波数ともに下げた場合を考える。

電圧値のみを下げた場合

電圧は実行時間には関係がなく、消費電力の変動が消費エネルギーに影響する。また、これまでの考察から、P_{static}、P_{static} ともに V との相関がある為 ($P_{dynamic} \propto V^2$、$P_{static} \propto$ V) 電圧値を下げれば、消費エネルギーも減少する。

動作周波数のみを下げた場合

動作周波数は実行時間に比例関係で影響する。また、消費電力のうち $P_{dynamic}$ のみが動

Vol.2010-ARC-192 No.2
Vol.2010-HPC-128 No.2
2010/12/16

表 7 性能比較 (演算性能 (GFLOPS) とバンド幅 (MB/sec)))

	Voltage	f:down	f:default	f:up
nbody (J)	default	103.116	100.014	97.796
	down	87.366	84.496	82.469
memcopy (mJ)	default	782.750	730.449	711.276
	down	681.694	638.936	624.713
bfs (mJ)	default	5996	5948	5685
	down	5362	5185	4983
dynproc (mJ)	default	10397	9796	9448
	down	9039	8578	8162
srad (J)	default	1335	1304	1260
	down	1169	1123	1082

作周波数と比例している為、実行時間による影響の方が大きくなり、動作周波数を下げると消費エネルギーは増加する。

6. 関 連 研 究

浅井ら[7] は、ＣＰＵ上において学習とＤＶＦＳを用いて消費電力削減を行った。パフォーマンスカウンタによって得た値と性能の相関を回帰分析によりあらかじめ学習させる。そして目的のプログラムを実行する際、一定間隔毎にパフォーマンスカウンタの値を受け取り、先に学習させた結果とあわせて次のインターバル実行間にてユーザが設定した性能を下回らない最低の動作周波数に変更して消費電力を抑えている。理論値では、性能を設定値を達成しつつ最大 24.9%の消費電力削減を達成している。

Callange ら[8] は、様々な世代の NVIDIA GPU におけるメモリ読み込みや演算命令、テクスチャーアクセスなどの消費電力について調査している。その結果、ＤＲＡＭを用いる場合よりもテクスチャーキャッシュを用いるほうがメモリ要求あたりの消費エネルギーは抑えられるとした。

Ma ら[9] は、消費電力とグラフィックスアプリケーションの相関関係について研究している。我々同様、パフォーマンスのプロファイルを用いて消費電力における統計的モデルを作っている。彼らは NVIDIA PerfKit と用いている。これはグラフィックスアプリケーション用のＧＰＵコンポーネントの利用を確認するものであり、例えば vertex や pixel shader の利用などである。彼らのプロファイルにはグローバルメモリアクセスに関するものは含まれない。しかし、我々の研究でグローバルメモリアクセスはＧＰＵのカーネル関数における消費電力に非常に大きく寄与していることが分かっている。

7. おわりに

7.1 ま と め

本研究では、実行プログラム毎に消費電力の大きさが異なることに着目し、プログラムの特徴と消費電力の関係性について解析した。プログラムの特徴としてはパフォーマンスカウンタ、解析手法としては線形回帰分析を用いて、?? という高精度な予測ができた。さらに、消費エネルギー削減の手法としてＣＰＵの省電力に広く用いられる「ＤＶＦＳ」に注目しＧＰＵにおける電圧値と動作周波数を変更しＧＰＵにおける消費電力の変化を調査した。その際に、電力測定の結果からＧＰＵでの消費電力は実行プログラムに関する電力 ($P_{dynamic}$) と関係しないそれ以外の電力 (P_{static}) という２つの要素に分けられると仮定しそれぞれが電圧値と動作周波数とどのような関係があるのかを解析した。その結果、$P_{dynamic}$ は電圧値の２乗と動作周波数に比例し、P_{static} は電圧値に比例していると推測できる結果を得た。

以上のことから、パフォーマンスカウンタの値から消費電力を予測し電圧値・動作周波数と消費電力の関係が分かった為、様々な環境でのＧＰＵにおける消費電力を高精度に推測することが可能となった。

7.2 今後の課題

今回の実験では、消費電力に注目しその値を正確に予測した。今後はその予測を用いＧＰＵプログラムによる消費エネルギー削減に取り組む必要がある。現状では電圧値と動作周波数にの変更のみによって消費電力の動向を追ったが、Baghsorkhi らによるカーネルの性能モデルを用いたＧＰＵ向け実行時間予測手法などを応用することで、ＧＰＵの電力性能の最適化に取り組む。

謝辞 本研究運の一部は科学技術振興機構戦略的想像研究推進事業「Ultra-Low-Power HPC:次世代テクノロジのモデル化・最適化による超低消費電力ハイパフォーマンスコンピューティング」によるものである。

参 考 文 献

1) L.Nyland,M.Harris,and J.Prins：”Fast N-body simulations with CUDA,”in GPU Gems 3,H.Ngyuen,Ed. Addison Wesley Professional,Augaut 2007,ch.31.
2) M.Schatz,C.Trapnell,A.Delcher,and A.Varshney,”High-throughput sequensce alignment using graphics processing units,”BMC Bioinformatics,vol.8,no.1,pp.474+,2007.[online]. Available:http://dx.doi.org/10.1186/1471-2105-8-474
3) S.S.Stone.J.P.Haldar,S.C.Tsao,W.Mei,Z.P.Liang,and B.P.Sutton,”Accelerrating ad-

Vol.2010-ARC-192 No.2
Vol.2010-HPC-128 No.2
2010/12/16

vanced mri reconstructions on gpus,"in Proceedings of the 5th conference on Computingfrontiers,2008,pp.261-272.[Online],Available:http://dx.doi.org/10.1145/1366230.1366276
4) T.Mudge,"Power:a first-class architectural design constraint,"Computer,vol.34,no.4,pp.52-58,2001.[Online].Available:http://dx.doi.org/101109/2.917539
5) S.Che,M.Boyer,J.Meng,D.Tarjan,J.W.Sheaffer,S.-HLee,and K.Skadron,"Rodinia:Abenchmark suite for heterogeneouus computing,"in Proceedings of the IEEE International Synposium on Workload Characterization(IISWC),2009,pp.pp.44-54
6) NVIDIA Corporation,"The CUDA Profiler",2009
7) 浅井雅司, 池田佳路, 佐々木宏, 近藤正章, 中村宏, 統計処理に基づくコンパイラ協調型ＤＶＦＳ手法. 情報処理学会論文誌, No.8,pp.43-48,2006
8) S.Collange,D.Defour,and A.Tisserand,"Power consumtion of gpus from a software perspective,"in Workshop on Using Emerging Parallel Architectures for Computational Science(in conjunction with ICCS'09).Springer,2009,vol.5544,ch.92,pp.914-923
9) X.Ma,M.Dong,L.Zhoneg,and Z.Deng,"Statical power conmption analysis and modeling for gpu-based computing,"in Workshop on Power Aware Computing and Systems(HotPower '09),2009)