

論文

DEGIMA における LINPACK 電力性能 評価

Energy Performance for LINPACK Benchmark on DEGIMA

概要

本論文では AMD/ATI Radeon HD 5870 を用いた GPU クラスタである DEGIMA における LINPACK ベンチマーク消費電力性能について評価結果を紹介する . HD 5870 GPU はプログラム実行中に電圧・動作周波数を調整することが可能である . DEGIMA では LINPACK ベンチマーク実行中に電圧・周波数の調整を行わなかった場合 1.4698 GFlops/Watt であったものが , 電圧・周波数の動的な調整を行った場合には 1.9658 GFlops/Watt まで向上することを確認した .

Abstract

GPU Computing has lately attracted for energy efficiency. Most of GPU computing system are using for coarse-grained optimization for power-consumption and not for energy efficiency. In this paper, we propose an fine-grained optimization method for enegy efficient GPU computing. We use AMD/ATI Radeon HD 5870 GPU system and introduce its power consumption model in relation between energy-efficiency(flops/watt) and system parameters such as GPU frequency and voltage. We implement an enegy controllable library with our power consumption model and apply it to the LINPACK benchmark. We found that the energy efficiency improved from 1.47 GFlops/Watt to 1.9658 GFlops/Watt using our method for LINPACK banchmark.

1. High Performance Computingの動向とGPU

High Performance Computing(HPC)の動向を表す指標としてTOP500プロジェクトが毎年2回発表するTOP500リストがある。TOP500リストとは、LINPACK¹⁾をベンチマークとして、世界中のスーパーコンピュータの性能を計測し、その性能によって半年に1度ランキングを行ったものをリストにしたものである。

1.1 国内の大学におけるHPCの動向

現在最新の2011年11月付TOP500リストには国内のシステムが30台ランクインしている²⁾。その中で、旧七帝大にある計算機センターは2システムがランクインしている。長崎大学のGPUクラスタシステムのDEGIMA(DESTINATION for GPU Intensive MACHINE)も含めたLINPACK benchmarkにおける実行性能(R_{max})と理論ピーク性能(R_{peak})を比較したのが図1である。^{*1}

長崎大学DEGIMAと図1の計算機センターのシステムとの大きな違いはDEGIMAはGPUをシステムの構成要素として用い、GPUの演算性能が全体の大部分を占めていることである。

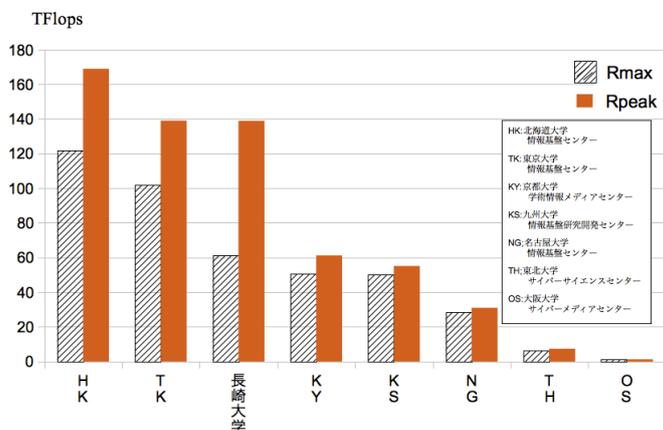


図1 TOP500 リストにおける旧七帝計算機センターと長崎大DEGIMAの性能比較

*1 比較対象に入っていないが K computer, Tsubame2, T2K-tsubakuba などより高い R_{max} , R_{peak} を持ったシステムが国内では他にランクインしている。

1.2 TOP500 における GPU 搭載マシン

近年消費電力効率の面で GPU を用いたクラスタマシンが世界的に使われはじめており、TOP500 リストの中にも GPU クラスタシステムの数が増加する傾向にある。例えば 2011 年 11 月の TOP500 リストでは、全体のうち 39 台（そのうち 2 台は AMD GPU、2 台は Cell、残りは Nvidia GPU³⁾）、国内だけでも 3 台の GPU クラスタシステムがランクインしている。長崎大学の GPU クラスタマシン DEGIMA もその 1 つである。この中で長崎大学が図 1 において低予算で TOP500 リストにランクインできているのは GPU を用いているためである。

1.3 Green500

Green500 とは TOP500 リストにランキングされたスパコンを消費電力性能 (Flops/W) でスパコンをランキング付けし直したものである。TOP500 と同じく約 6ヶ月に 1 度ランキングを発表している。図 2 では 2011 年 11 月にランキングされた上位 50 位を順位順に示している⁴⁾。この中の 60% は GPU クラスタマシンであり、また上位に多数ランクインしていることから電力性能の良さがわかる。GPU は高密度に浮動小数点演算器を実装しているため、演算器あたりのトランジスタ量が小さくなる傾向にあり、演算性能あたりの消費電力量が小さくなるような設計になっている。そのため Green500 では GPU を用いたシステムが上位に位置する傾向となっている。

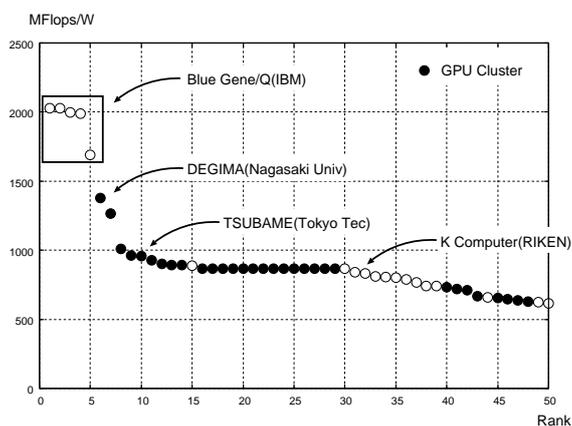


図 2 2011 年 11 月における Green500 リスト。上位 50 位までの順位 (横軸) と電力性能 (縦軸) を表している。また GPU クラスタシステム (黒丸) とその他のシステム (白丸) とに区別して表している。

1.4 エクサフロップスにおける消費電力の問題

2011年11月に理化学研究所のK-computerが世界初のLINPACK benchmark 性能 10 PFlops(ペタフロップス) を達成した。また将来的にはその100倍の1秒間に100京回の浮動小数点演算を実行可能なエクサフロップスシステムが登場するであろうと予測されている。図3は、過去のTOP500リストの動向を示している。これまでの動向からエクサフロップスのシステムは2019年前後に登場することが予測できる。

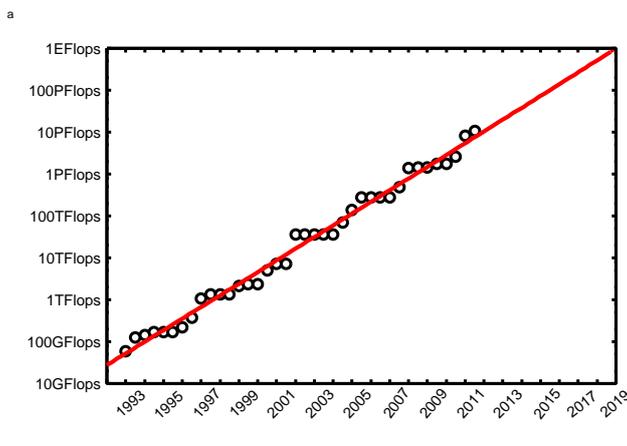


図3 TOP500における演算性能の推移．縦軸がTOP500#1の R_{max} 性能，横軸が年代順を表している．赤は2019年までの外挿を表す⁵⁾

現在TOP500で第1位である理化学研究所の計算科学研究機構のK-computerはLINPACK実行時の演算性能は11.28PFlops，消費電力は12.66 MWである。消費電力あたりの演算性能値，すなわちFlops/Wで表される消費電力性能が現在のままで一定であると仮定すると，エクサフロップスのシステムではK-computerの100倍の1266 MWもの電力を必要とすることになり，これは福島第一原子力発電所の1号機における発電能力の約2.75倍にも相当する莫大な電力となる⁶⁾。

図4はこれまでのGreen500の動向を表している。この図から消費電力性能の向上率は4年で5.67倍程度であることがわかる。この向上率でいけば2019年には現在の32倍程度の電力性能になることが予想され，その頃に登場するエクサフロップスの計算機が消費する電力は約39.38 MW (= 12.66 MW × 100/32.149) も必要になることを見積もることができる。これは，仮にこれまで通り順調に

消費電力性能の向上が続いたとしても、エクサフロップシステムを稼働させるのに必要な電力は現在の K-computer の 3 倍以上も必要になることを意味している。

そのため、消費電力性能の向上のための研究は今後ますます重要になる。

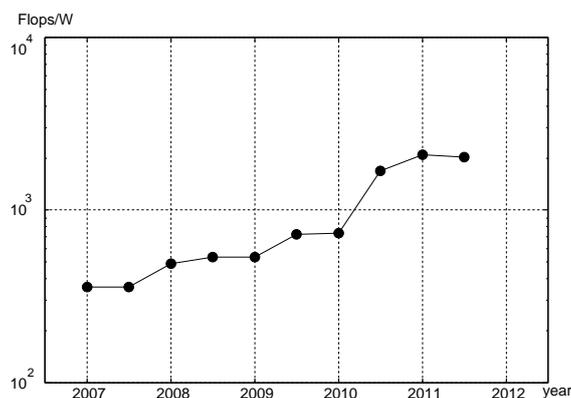


図4 Green500 No.1 の推移

2. AMD/ATI Radeon 5870

DEGIMA で用いる AMD/ATI Radeon 5870 GPU の仕様および電力制御機構について説明する。

2.1 AMD/ATI Radeon 5870 の仕様

まず、AMD/ATI Radeon 5870 の仕様を表 1 に示す⁷⁾。

Engine clock speed	850 MHz
Processing power(単精度)	2.72 TFlops
Processing power(倍精度)	544 GFlops
Memory clock speed	1.2GHz
Memory bandwidth	153.6 GHz

表1 ATI Radeon 5870 の仕様

次に表 2 で AMD/ATI Radeon 5870 と他の GPU を比較した。ただしこの表で倍精度のピーク性能が最も高い Tesla M2090 は、Radeon5870 に比べて価格が 17 倍ほど高く、価格性能比では Radeon 5870 は Tesla M2090 よりも 14 倍良

いことがわかる .

項目	Radeon 5870	Tesla C2070	Tesla M2090	GTX 580
Process Technology	40(nm)	40(nm)	40(nm)	40(nm)
理論性能 (単精度)	2.72TFlops	1.03TFlops	1.33TFlops	1.581TFlops
理論性能 (倍精度)	544GFlops	515GFlops	665GFlops	198GFlops
熱設計電力	188W	215W	225W	244W
価格 (2011 年 12 月現在)	25983 円	212746 円	452025 円	39580 円
名目価格性能 (倍精度)	20.94MFlops/円	2.42MFlops/円	1.45MFlops/円	5.00MFlops/円
名目消費電力性能 (倍精度)	2.894GFlops/W	2.395GFlops/W	2.956GFlops/W	0.81GFlops/W

表 2 同一製造プロセスでの各社 GPU 製品ごとの諸元比較. 理論演算性能, 熱設計電力 (TDP: Thermal Design Power), 2011 年 12 月現在の市場価格などを比較している ⁸⁾⁹⁾¹⁰⁾¹¹⁾¹²⁾¹³⁾¹⁴⁾

2.2 AMD GPU における 3 つの動作レベル

AMD/ATI GPU には計算負荷によって自動的に切り替わる 3 つの動作レベルというものがある . これは各レベルに対し , GPU のエンジン周波数 , メモリ周波数 , 駆動電圧の 3 つ , 合計 9 つのパラメータがあらかじめ設定されている . 表 3 がこれら 3 つのレベルのパラメータ値を表したものである .

また , 表 3 の 9 つのパラメータ値は AMD Display Library(ADL) を使用することでユーザプログラムから変更することが可能である . AMD Display Library(ADL) とは , AMD/ATI 製 GPU のディスプレイドライバ諸機能にアクセスするための C 言語プログラミングインタフェースである ¹⁵⁾ . AMD Radeon 5870 で ADL を用いた場合 , GPU のエンジン周波数を 80 MHz ~ 1200 MHz の範囲内で 5 MHz 刻みに , GPU のメモリ周波数を 150 MHz ~ 1400 MHz の範囲内で 5 MHz 刻みに , GPU の駆動電圧を 1.062 V ~ 1.212 V の範囲内で 5 mV 刻みに変更可能である . また表 3 のパラメータの他にも GPU の温度や移動平均負荷などの各種情報を取得することも可能である .

2.3 AMD GPU における粗粒度電力制御機構

AMD Radeon 5870 GPU は負荷や温度を監視することでさきほどの動作レベルを動的に切り替える電力制御機能を備えている . その電力制御機能が実際のプログラムを実行させた場合にどのように動作しているのかを実際に観測して

Level	Engine frequency(MHz)	Memory frequency(MHz)	Core voltage(V)
2	850	1200	1.212
1	600	900	1.112
0	157	300	1.062

表3 ATI Radeon 5870 の3つの動作レベルにおける各標準パラメータ値．レベル0, 1, 2それぞれでGPUのエンジン周波数, メモリ周波数, 駆動電圧をパラメータに持つ．

みることにした．

図5は, LINPACK ベンチマーク実行中に Active Percent 値, GPU 温度をADL 経由で取得した結果, および, プログラムがGPUを計算に用いている期間を表したものである．また図5は実際のLINPACK 実行時間のうちの一部(25秒)だけに注目して表示したものである．

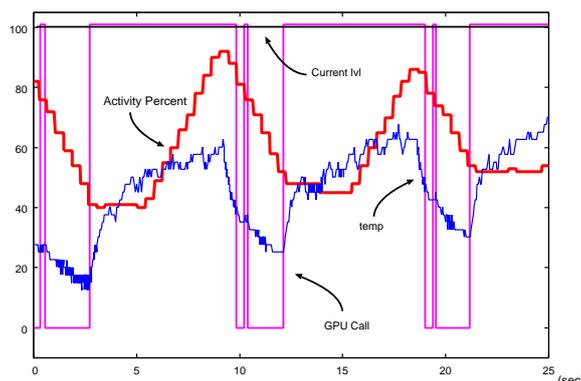


図5 LINPACK 計算中におけるRadeon 5870 GPU 標準の粗粒度動的消費電力制御の観測結果．GPUの移動平均負荷(Active Percent), 温度(temp), 動作レベル(Current lvl)と, LINPACK ベンチマーク中でのGPUの実行状況(GPU Call)の推移．全計算のうちの一部(25秒のみ)を取り出して表示している．

図5はLINPACK 計測中でGPU 負荷が比較的高い状況の25秒間程度を抜粋したのものである．図5の黒色, 青色, 赤色, ピンク色の測定値はそれぞれ, GPU 動作レベル(Current lvl), GPU 温度(temp), GPU 移動平均負荷(Active Percent), GPU の呼び出し状況(GPU Call)を表す．

LINPACK ベンチマーク実行中にはホスト計算機は計算をGPUにロードし

たり，GPU を使わず CPU のみで計算を行ったりといったことを繰り返すため，図 5 のピンク色で示した GPU の呼び出し状況 (GPU Call) は ON/OFF を繰り返していることがわかる．図 5 の青色で示した GPU 温度 (temp) は，GPU が Call された直後から上昇していることがわかる．GPU Call が終了する直前に GPU 温度はピークから低下し始めているのは，GPU Call の終了間際に計算が終了し GPU とホスト計算機の通信が始まっているためである．言い換えると図 5 の GPU が Call されている期間 (図では GPU Call の値を 100 として表示) に GPU-ホスト間通信を含めているためである．

問題は，黒色で示した GPU 動作レベル (Current lvl) である．図 5 で示した LINPACK 計測 25 秒間に動作レベルは値 2 で一定，つまり全く変化していないことがわかる．実際には LINPACK ベンチマークを実行直後に動作レベルは 0 から 2 に変化し，LINPACK ベンチマークが完了直後に動作レベルは 2 から 0 に変化していることを観測した．このように AMD GPU 付属の電力制御機能は非常に粗い粒度でしか電力制御を行っていないことがわかる．

図 6 に本研究の評価マシンにおけるアイドル状態と実際に LINPACK ベンチマークを計算させた時の高負荷，低負荷の状態の消費電力の値をそれぞれ示す．このときデフォルトの値を赤で，ADL によってパラメータを変えた時の値を青で示す．ここでデフォルトとは，Radeon 5870 で動作レベル 2 で設定しているパラメータのことを表す．また ADL によって変化させたパラメータは本研究で最も良い消費電力性能を得たパラメータである．

ここで図 6 より，デフォルトの状態からパラメータを変えた状態ではアイドルの場合で約 37.4%，LINPACK 計算時における高負荷のときで約 21.4%，低負荷のときで 27.7% も消費電力が削減出来ることが解る．

これを元にし，図 5 で示した粗粒度な電力制御を，細粒度にしたイメージ図を図 7 で表す．ここでパラメータを変えたことによって計算速度は変わらないものとする．

以上より先ほどの図 7 のように行うことが出来れば最大で約 18.4% もの消費電力を削減出来ると推測される．

3. 提案手法

前章で AMD/ATI GPU 付属の電力制御は非常に粗粒度であることがわかった．そこで本章では非常に粗粒度である AMD/ATI GPU 付属の電力制御機構を

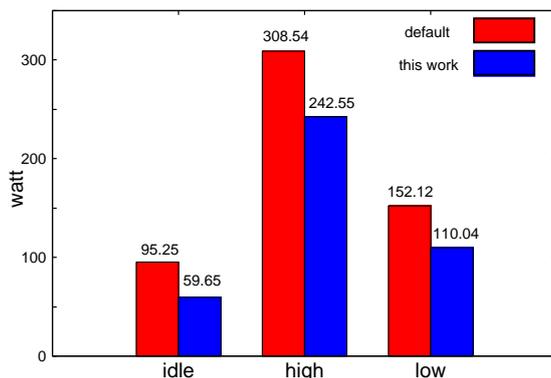


図6 Radeon 5870 GPU の消費電力の値を表している．デフォルトを赤で，パラメータを変えた時を青で表し，左からアイドル状態 (idle)，LINPACK 計算時の高負荷状態 (high)，LINPACK 計算時の低負荷状態 (low) を示している．

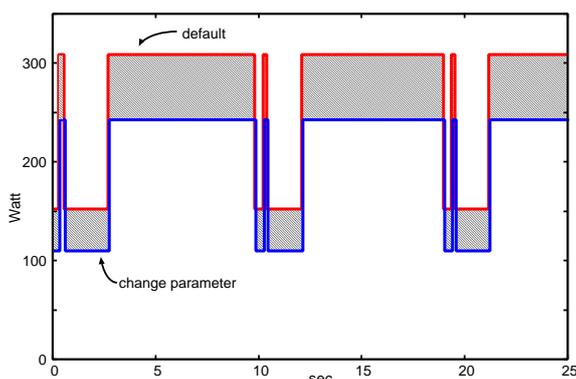


図7 図5におけるデフォルト (赤) とパラメータを変えた時 (青) それぞれの消費電力のイメージ図を 25 秒抜き出して表している．またそれによって削減される消費電力を斜線で示している．

改善するために，本研究で用いる提案手法を示す．

3.1 提案手法

前章で，AMD/ATI GPU 付属の電力制御機構は非常に粗粒度であり改善の余地が残されていることがわかった．そこで今回我々はこの電力制御の粒度をより細かく行うことで電力性能向上を実現しようと考えた．

実現方法として大雑把には，電力制御を行う API ライブラリを通しアプリケーションプログラム内で電力制御を行う API 方式，または GPU のドライバから得られる情報を取得しつつながら電力制御を行うアプリケーションとは独立して動作するようなサーバプログラム型制御方式の 2 つの方法を考えた．本

提案ではアプリケーションプログラム中でどのタイミングで GPU が Call されるのかがあらかじめ判別可能であることを前提とした前者の API 方式を採用することにした。後者のサーバ方式については，GPU がいつ呼ばれたのかを推測する方法が自明ではなかったため将来の課題とすることにした。

3.2 電力制御 API

今回開発する電力制御 API ライブラリは，(1) 消費電力を上げる，(2) 消費電力を下げる，の 2 種類のみをユーザが指定するようなインターフェースを持ち，電力を上下させるために必要な動作周波数や駆動電圧といったシステムパラメータはライブラリ自身が自動的に決定するような設計を行った。図 8 に，上記の 2 つの API を C 言語サンプルコードで表した。EL_SetHighestAutomatic が消費電力を上げる API，EL_SetLowestAutomatic が消費電力を下げる API である。EL_DEVICE は API で用いるデバイスの種類を表し，将来的には Radeon 5870 以外のデバイスにも対応できるようにした。補足であるが，EL_Init はデバイスごとに必要になるライブラリ内部状態の初期設定を行うための API である。

```

int main()
{
  EL_DEVICE dev = EL_Init(EN_DEVICE_TYPE_HD5870); // EnergyLibrary Initialization
  .
  .
  .
  EL_SetHighestAutomatic(dev); // EnergyLibrary API
  .
  .
  .
  EL_SetLowestAutomatic(dev); // EnergyLibrary API
  .
  .
  .
}

```

The diagram illustrates the execution flow in the provided C code. It shows three vertical sections: 'host part' at the top, 'GPU part' in the middle, and 'host part' at the bottom. The 'host part' contains the initialization and the two API calls. The 'GPU part' is indicated by a double-headed arrow between the two API calls, signifying that the GPU is active during this period.

図 8 開発した電力制御 API。電力制御 API は C 言語ライブラリとして実装。電力制御 API はホストコードで用い，GPU の計算を開始直前と終了直後の 2 箇所のみで利用する。

EL_SetLowestAutomatic は GPU が計算を行わないときに呼び出すため，GPU が最も電力を消費しない各パラメータの最低値を設定する。また EL_SetHighestAutomatic は主に GPU が計算を行うときに呼び出す。ここで表 3 の値が最適値でない理由については後述する。

4. LINPACK ベンチマークにおける消費電力性能の測定

今回開発する電力制御 API ライブラリは，電力を上下させるために必要な GPU エンジン動作周波数や GPU メモリ動作周波数，駆動電圧といったシステムパラメータはライブラリ自身が自動的に決定しなければならない．

そこで本章では，実際のアプリケーションで消費電力性能と動作周波数・駆動電圧の関係がどうなっているのかを実測してみることにした．実測には DGEMM と LINPACK ベンチマークの 2 種類を用いて行った．DGEMM(Double-precision General Matrix Multiply) とは LINPACK ベンチマーク内で用いる倍精度行列乗算のことである．

4.1 評価環境

我々が評価に用いた環境は図 9 の通りである．模式図で示したデジタルマルチメータは写真の上部に，ホスト計算機は写真の下部になっている．また AMD Radeon 5870 GPU はホスト計算機の中央付近にある．

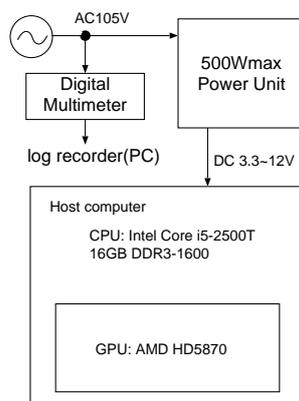


図 9 評価環境模式図．

4.2 DGEMM の測定結果

実際に DGEMM を用いて GPU の性能を測定する．DGEMM の行列の大きさを $N=42000$ と定め，GPU の駆動電圧を設定出来る範囲 $1.062V \sim 1.212(V)$ における最小値，中間値，最大値である $V=1.062(V)$ ， $V=1.137(V)$ ， $V=1.212(V)$ の 3 つで計算を行う

4.2.1 演算性能

GPU のエンジン周波数とメモリ周波数をそれぞれ変えることで，演算性能の

比較を行った．これを図 10 に示す．この図からメモリ周波数を一定にしエンジン周波数を上げると，演算性能が下がるという奇妙な現象を発見した．例えば，メモリ周波数 750MHz のときエンジン周波数を 700MHz から 850MHz に上げていくと，演算性能は上がると思われるが，実際に図 10 では下がっている．そのため演算性能を上げるにはエンジン周波数とメモリ周波数の比を考える必要がある．また GPU の駆動電圧は演算性能にあまり影響を及ぼさないことが解る．

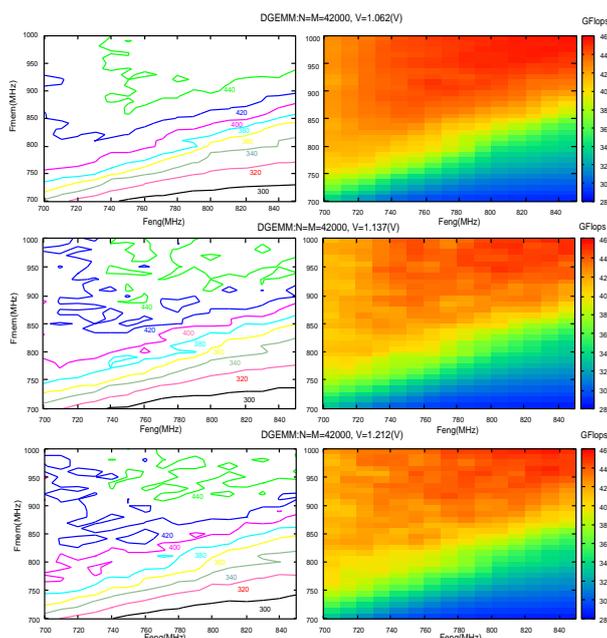


図 10 DGEMM 演算性能測定結果．DGEMM N=42000 における GPU のエンジン周波数 (横軸)，メモリ周波数 (縦軸)，GPU の駆動電圧 (上から V=1.062(V)，V=1.137(V)，V=1.212(V)) と演算性能 (GFlops) の関係と比較したものの等高線表示 (左) とカラーマップ (右)

4.2.2 消費電力

GPU のエンジン周波数とメモリ周波数をそれぞれ変えることで，デジタルマルチメータによって得られた消費電力の比較を行った．これを図 11 に示す．この図から消費電力は GPU のエンジン周波数とメモリ周波数が高ければ高いほど大きくなる．また GPU の駆動電圧が大きければ大きいほど消費電力が大きくなることを解る．

4.2.3 消費電力性能

得られた演算性能と消費電力を用いて消費電力性能の値を導く．これによ

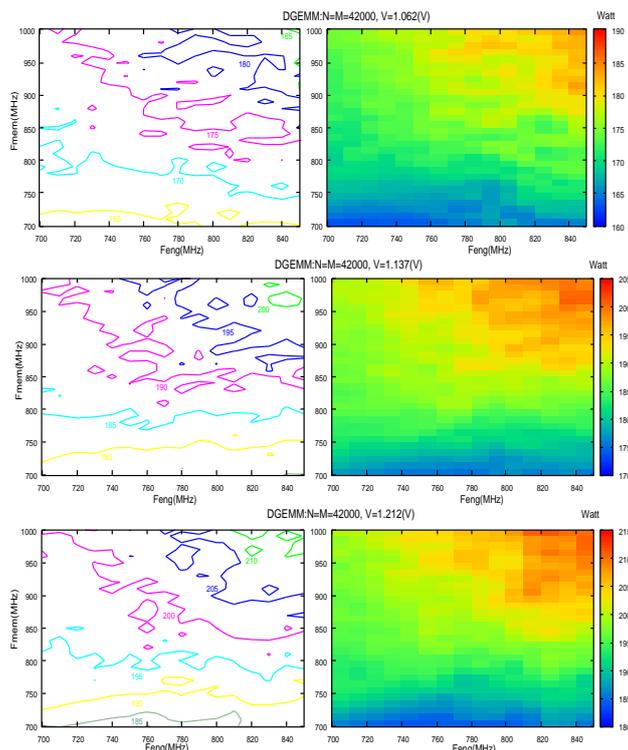


図 11 DGEMM 消費電力測定結果．DGEMM N=42000 における GPU のエンジン周波数 (横軸)，メモリ周波数 (縦軸)，GPU の駆動電圧 (上から V=1.062(V)，V=1.137(V)，V=1.212(V)) と消費電力 (Watt) の関係を比較したものの等高線表示 (左) とカラーマップ (右)

て得られた消費電力性能の比較を図 12 に示す．消費電力性能は演算性能と同じく，GPU のエンジン周波数とメモリ周波数の関係に奇妙な現象が見られる．そのため GPU のエンジン周波数とメモリ周波数を上げてても，消費電力性能が良くなるとは限らないことが解る．また GPU の駆動電圧を大きくすると消費電力は上がるが，演算性能は変わらないため，消費電力性能は小さくなる．このとき最大の消費電力性能が得られたパラメータはエンジン周波数 770MHz，メモリ周波数 950MHz，駆動電圧 1.062V であった．

4.3 LINPACK の測定結果

実際に LINPACK を用いて GPU の性能を測定する．LINPACK の行列の大きさを N=39680，ブロックサイズを NB=1280 と定め，GPU の駆動電圧を V=1.062(V) として計算を行う．

4.3.1 演算性能

GPU のエンジン周波数とメモリ周波数をそれぞれ変えることで，演算性能の

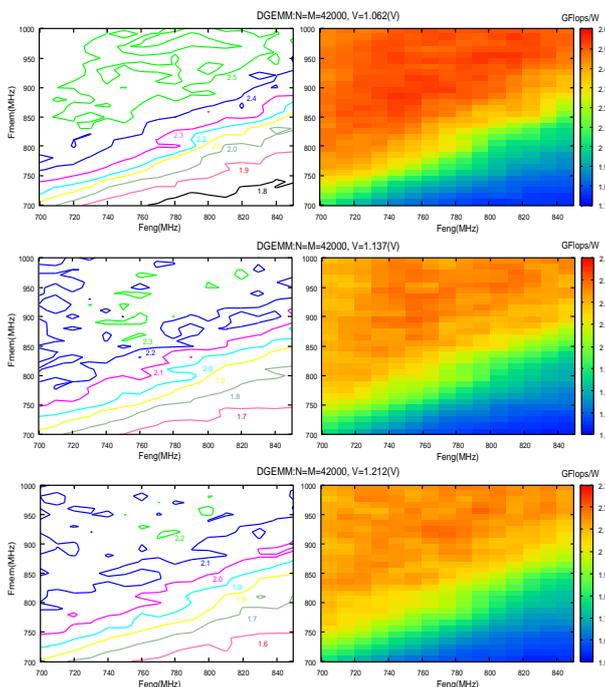


図 12 DGEMM 電力性能測定結果．DGEMM N=42000 における GPU のエンジン周波数 (横軸)，メモリ周波数 (縦軸)，GPU の駆動電圧 (上から $V=1.062(V)$ ， $V=1.137(V)$ ， $V=1.212(V)$) と消費電力性能 (GFlops/W) の関係を比較したものの等高線表示 (左) とカラーマップ (右)

比較を行った．これを図 13 に示す．この図から DGEMM の演算性能と同じく GPU のエンジン周波数とメモリ周波数の関係に奇妙な現象がみられる．また DGEMM と LINPACK における演算性能は GPU のエンジン周波数，メモリ周波数と演算性能の関係の結果から非常に似ていることが解る．

4.3.2 消費電力

GPU のエンジン周波数とメモリ周波数をそれぞれ変えることで，デジタルマルチメータによって得られた消費電力の比較を行った．これを図 14 に示す．この図から DGEMM の消費電力と同じく GPU のエンジン周波数とメモリ周波数が高ければ高いほど大きくなる．また DGEMM と LINPACK における消費電力は GPU のエンジン周波数，メモリ周波数と消費電力の関係の結果から非常に似ていることが解る．

4.3.3 消費電力性能

得られた演算性能と消費電力を用いて消費電力性能の値を導く．これによって得られた消費電力性能の比較を図 15 に示す．DGEMM の消費電力性能と同

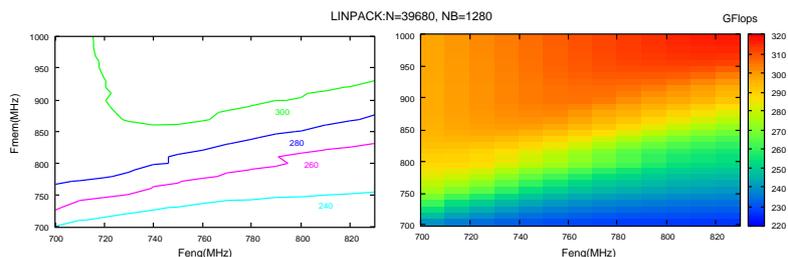


図 13 LINPACK 演算性能測定結果．LINPACK N=39680，NB=1280，GPU の駆動電圧 V=1.062(V) における GPU のエンジン周波数 (横軸)，メモリ周波数 (縦軸) と演算性能 (GFlops) の関係の等高線表示 (左) とカラーマップ (右)

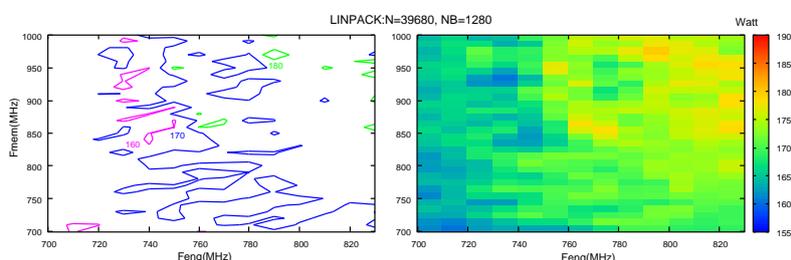


図 14 LINPACK 消費電力測定結果．LINPACK N=39680，NB=1280，GPU の駆動電圧 V=1.062(V) における GPU のエンジン周波数 (横軸)，メモリ周波数 (縦軸) と消費電力 (Watt) の関係の等高線表示 (左) とカラーマップ (右)

じく，GPU のエンジン周波数とメモリ周波数の関係に奇妙な現象がみられる．そのため GPU のエンジン周波数とメモリ周波数を上げてても，消費電力性能が良くなるとは限らないことが解る．また DGEMM と LINPACK における消費電力性能は GPU のエンジン周波数，メモリ周波数と消費電力性能の関係から非常に似ていることが解る．このとき最大の消費電力性能が得られたパラメータはエンジン周波数 770MHz，メモリ周波数 950MHz，駆動電圧 1.062V であった．

5. 電力性能モデル

ここまでは DGEMM 実測に基づくパラメータを用いて電力制御を行ったが，より最適な電力制御パラメータが存在する可能性もあるので適当な電力性能モデルを作成し評価を行った．

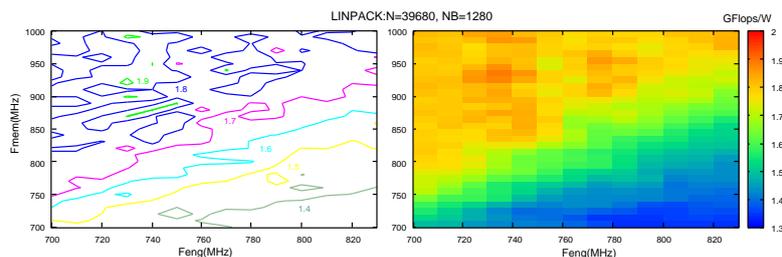


図 15 LINPACK 電力性能測定結果．LINPACK N=39680，NB=1280，GPU の駆動電圧 $V=1.062(V)$ における GPU のエンジン周波数 (横軸)，メモリ周波数 (縦軸) と消費電力性能 (GFlops/W) の関係の等高線表示 (左) とカラーマップ (右)

5.1 電力性能モデルの種類

ここでは電力性能モデルを 2 種類導入する．一つ目は DGEMM の消費電力性能結果を直接適当な関数にあてはめたモデル化を用意する (モデル 1)．もう一つ目は DGEMM の演算性能結果，消費電力結果ごとにそれぞれ同様にモデルを作成しそれらを除算したものを用意する (モデル 2)．

5.2 電力性能モデル 1

モデル 1 では図 12 によって得られた消費電力性能の値から式 1 とした．図 16 はこれを図示したものである．このとき最大の消費電力性能が得られたパラメータはエンジン周波数 765MHz，メモリ周波数 930MHz，駆動電圧 1.062V であることがわかった．

$$\begin{aligned}
 f(f_{eng}, f_{mem}) = & -1.73659 \times 10^{-18} f_{eng}^4 + 1.0627 \times 10^{-18} f_{mem}^4 - 4.38584 \times 10^{-18} f_{eng}^3 f_{mem} \\
 & -1.24579 \times 10^{-17} f_{eng} f_{mem}^3 + 1.81337 \times 10^{-17} f_{eng}^2 f_{mem}^2 + 9.99745 \times 10^{-13} f_{eng}^2 f_{mem} \\
 & + 5.95166 \times 10^{-13} f_{mem}^3 - 2.1966 \times 10^{-12} f_{eng}^2 f_{mem} + 4.48017 \times 10^{-13} f_{eng} f_{mem}^2 \\
 & - 2.96352 \times 10^{-8} f_{eng}^2 - 9.68834 \times 10^{-8} f_{mem}^2 + 1.39058 \times 10^{-7} f_{eng} f_{mem} \\
 & - 2.27374 \times 10^{-3} f_{eng} + 1.95304 \times 10^{-3} f_{mem} + 9.80607 \times 10^{-1}
 \end{aligned} \quad (1)$$

5.3 電力性能モデル 2

モデル 2 ではモデル 1 と違い，演算性能と消費電力をそれぞれフィッティングさせたものを使い，消費電力性能のモデルを作成する．そこで図 17 をモデル図として考える．

またモデルにおける電力のパラメータをそれぞれ式 2 から式 6 で表す．ここでエンジン周波数の電力 W_{eng} ，メモリ周波数の電力 W_{mem} とともに，それぞれの周波数 f と電圧 V の 2 乗，係数の積とする．このとき， W_{host} は定数とする．こ

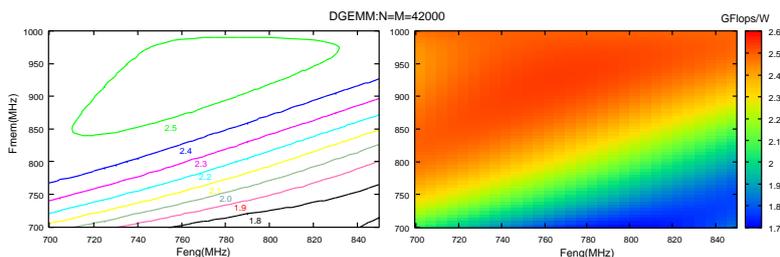


図 16 電力性能モデル 1 . DGEMM N=42000 の消費電力性能値 (図 12) から算出 .

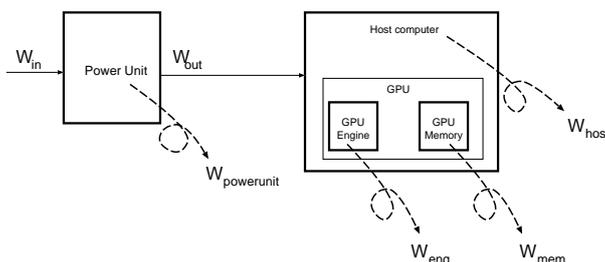


図 17 電力性能モデルの概念図 .

の理由として、今回変更するパラメータは、GPU のエンジン周波数とメモリ周波数であり、ホストのパラメータには一切変更をしない . そのため、GPU のパラメータ変更時に比べて、誤差は小さいと考えた . そのため式 4, 5, 6 はそれぞれスタティック消費電力を考えず簡略化することにした . また図 10 から GPU の駆動電圧は演算性能に影響を及ぼさないことが解った . そのため GPU のエンジン周波数とメモリ周波数のパラメータのみの関数系である式 7 で表す . また消費電力性能を式 8 で表す .

$$W_{powerunit} = W_{in} - W_{out} \quad (2)$$

$$W_{out} = W_{eng} + W_{mem} + W_{host} \quad (3)$$

$$W_{eng} = k_{eng} f_{eng} V^2 \quad (4)$$

$$W_{mem} = k_{mem} f_{mem} V^2 \quad (5)$$

$$W_{host} = Const \quad (6)$$

$$S(\vec{f}) = S(f_{eng}, f_{mem}) \quad (7)$$

$$E = \frac{S(\vec{f})}{W_{in}(\vec{f}, V)} \quad (8)$$

5.3.1 演算性能のフィッティング

次にモデル 2 であるが、モデル 1 同様図 10 から演算性能モデル、電力モデル

をそれぞれ同様にそれぞれ式 9 , 式 10 のように導出しそれらを除算することで導出した . これら図示したものを図 18 , 図 20 , 図 21 , に示す . またモデルと実測値との相対誤差を図 22 にそれぞれ示す . 最大の消費電力性能が得られたパラメータはエンジン周波数 765MHz , メモリ周波数 920MHz , 駆動電圧 1.062V であることがわかる .

$$\begin{aligned}
 f(f_{eng}, f_{mem}) = & -2.28649 \times 10^{-16} f_{eng}^4 + 1.9017 \times 10^{-16} f_{mem}^4 - 5.86035 \times 10^{-16} f_{eng}^3 f_{mem} \\
 & -2.34738 \times 10^{-15} f_{eng} f_{mem}^3 + 3.14867 \times 10^{-15} f_{eng}^2 f_{mem}^2 + 1.39896 \times 10^{-10} f_{eng} f_{mem}^2 \\
 & + 1.15289 \times 10^{-10} f_{mem}^3 - 4.23504 \times 10^{-10} f_{eng}^2 f_{mem} + 1.25702 \times 10^{-10} f_{eng} f_{mem} \\
 & + 1.64027 \times 10^{-7} f_{eng}^2 - 2.02001 \times 10^{-5} f_{mem}^2 + 2.34208 \times 10^{-5} f_{eng} f_{mem} \\
 & - 6.31176 f_{eng} \times 10^{-1} + 5.47227 f_{mem} \times 10^{-1} - 1.99416 \times 10^{-1}
 \end{aligned} \tag{9}$$

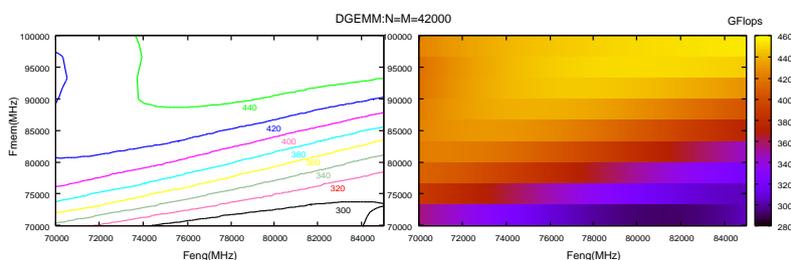


図 18 演算性能モデル . (DGEMM N=42000)

$$W_{out} = 3.60009 \times 10^{-4} f_{eng} + 4.33553 \times 10^{-4} f_{mem} + 89.9443 \tag{10}$$

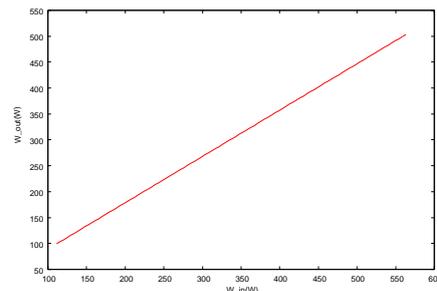
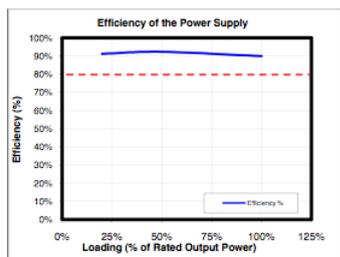


図 19 電源ユニット変換効率 . 今回用いる電源ユニットの変換効率 (左)¹⁶⁾ とそこから求められる W_{in} と W_{out} の関係 .

6. 結果

図 12 から , DGEMM の実測値における最大の消費電力性能が得られる値は

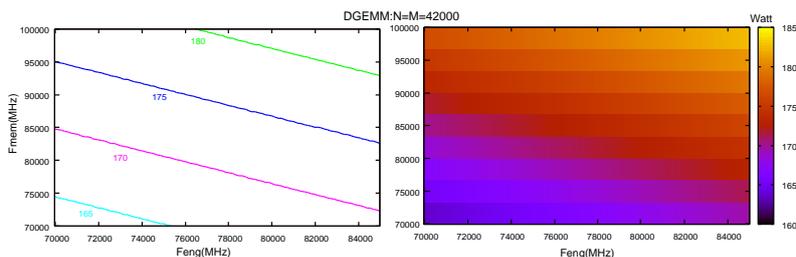


図 20 消費電力モデル . DGEMM N=42000 の消費電力の値 (図 11) を式 3 に関数フィッティングさせたものの等高線表示 (左) とカラーマップ (右)

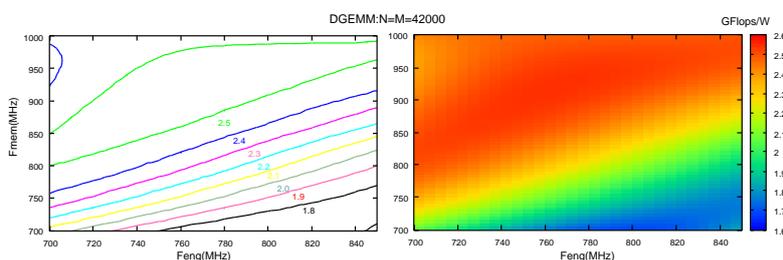


図 21 電力性能モデル 2 . DGEMM N=42000 における消費電力性能の値を図 18 と図 20 のモデルから計算したものの等高線表示 (左) とカラーマップ (右)

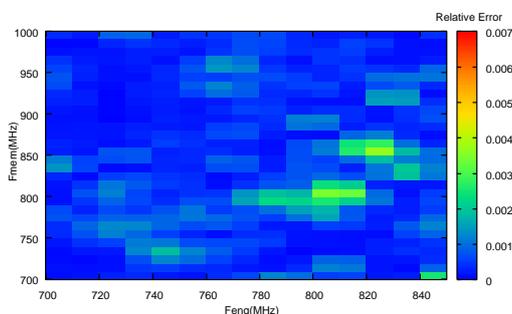


図 22 図 16 の電力性能モデルと実測値との相対誤差 .

エンジン周波数 770MHz , メモリ周波数 950MHz , 駆動電圧 1.062V であった . 本章ではこの値を用いて実際に LINPACK の消費電力性能を測定し , 本研究の有用性を確認する .

6.1 メモリ周波数の比較

図 23 では , LINPACK ベンチマーク計算時における GPU のメモリ周波数が , AMD/ATI Radeon5870 のデフォルトの状態と提案手法を用いたときで , どのように変化するかを表したものである . 本研究の提案手法が , デフォルトの状

態に比べてメモリ周波数が使用状況によって細かく制御出来ていることが解る .

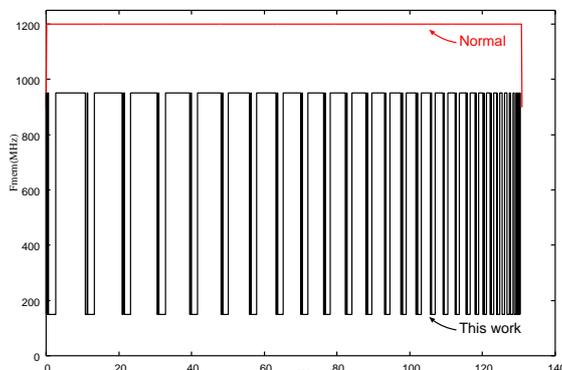


図 23 AMD/ATI Radeon 5870 の標準粗粒度電力制御機構 (Normal) と本提案の細粒度電力制御機構 (This work) における計算実行中の GPU メモリ周波数の推移 .

6.2 消費電力の比較

AMD/ATI Radeon5870 のデフォルトの状態と提案手法を用いたときでどのように電力が変化するかを表す . 図 24 では , 左側が LINPACK ベンチマークにおける電力 , 右側が積分値を示し , デフォルトの電力と提案手法を用いたときの電力の差をそれぞれの下に示している . このときデフォルトの状態では 220.972W , 提案手法を用いた場合では 159.552W の消費電力であり , 本研究の提案手法がデフォルトの状態に比べ , 消費電力が約 27.8% 減少していることが解る .

電力性能モデルの種類	エンジン周波数	メモリ周波数	消費電力
実測値	770MHz	950MHz	1.062V
モデル 1	765MHz	930MHz	1.062V
モデル 2	765MHz	920MHz	1.062V

表 4 実測値 (図 12) , 及びモデル式 (図 16 , 図 21) それぞれにおける極値の詳細 .

6.3 消費電力性能の比較

DGEMM の実測値 , モデル 1 , モデル 2 のそれぞれにおける最大の消費電力性能が得られるエンジン周波数 , メモリ周波数 , 駆動電圧の値を表 4 に示す . 表

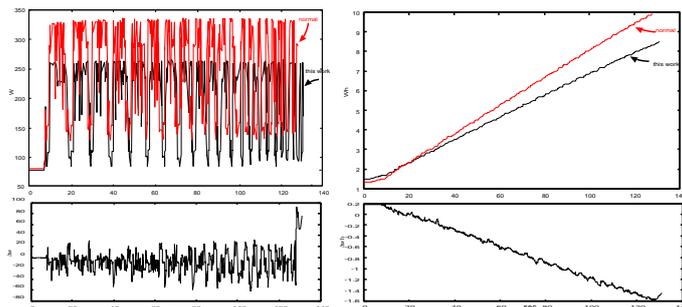


図 24 LINPACK ベンチマーク実行中の消費電力変動．AMD に標準付属の粗粒度動的電力制御機構を用いた場合 (normal) と，本提案の細粒度動的電力制御機構を用いた場合 (this work) との比較．下は制御方式の差 (this work - normal) を表す．

5, 図 25 では，提案手法を適用しないとき，また提案手法を表 4 のパラメータそれぞれに適用した LINPACK 性能，消費電力，電力性能の比較をまとめた．提案手法を用いた場合では，AMD/ATI Radeon5870 が計算時に使うデフォルトのエンジン周波数とメモリ周波数に比べると，低い値を使っているため LINPACK の演算性能は下回る．しかし提案手法の場合，細粒度に消費電力制御を行っているため，消費電力の削減に成功した．結果としてデフォルトの状態では電力性能が 1,4698Gflops/W だったものが，提案手法を適用した中で最も良い値であるモデル 2 の時は 1.9658Gflops/W となりデフォルトの状態よりも 33.7% の向上に成功した．また提案手法を適用する前に比べて提案手法を適用したどのパラメータも消費電力性能が良くなっているのが解る．

提案手法の有無	モデルの種類	LINPACK 性能	消費電力	消費電力性能
無	デフォルト	324.8 Gflops	220.972 W	1.4698 Gflops/W
有	実測値	310.6 Gflops	159.552 W	1.9472 Gflops/W
有	モデル 1	309.1 Gflops	162.794 W	1.8993 Gflops/W
有	モデル 2	308.6 Gflops	157.017 W	1.9658 Gflops/W

表 5 LINPACK ベンチマーク性能測定結果. 提案手法である細粒度動的消費電力制御手法の有無で比較 (N=39680, NB=1280).

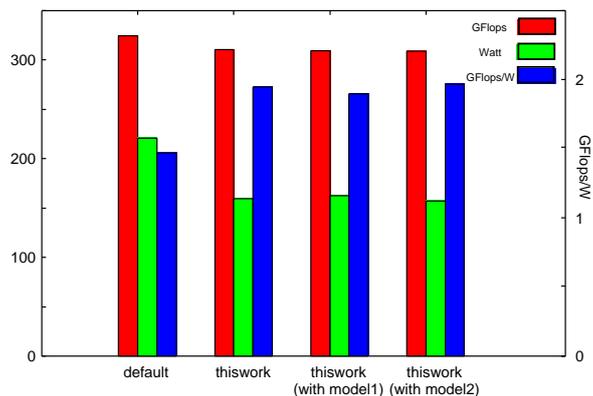


図 25 LINPACK ベンチマーク性能測定結果. 提案手法である細粒度動的消費電力制御手法の有無で比較 (N=39680, NB=1280). 演算性能 (GFlops) を赤, 消費電力 (Watt) を緑, 消費電力性能 (GFlops/W) を青で示す.

7. ま と め

今回我々は細粒度消費電力制御機構を用いプログラム実行中に GPU のエンジン周波数, メモリ周波数および電圧を動的に調整しながら LINPACK ベンチマークを実行した. 調整に用いるパラメータは, DGEMM を実測した場合と DGEMM 実測による電力モデル 2 種類の計 3 種類を用いて評価した. DGEMM 電力性能モデルを用いた場合の LINPACK 性能は電圧・周波数を全く調整しない場合に比べて, 電力性能が 1.4698 Gflops/W から 1.9658 Gflops/W へ向上することが確認できた. 今後はこれらの最適なパラメータを用いることで 2011 年 6 月および 2011 年 11 月時点での Green500 よりも高い電力性能が得られるであろうと考えている.

References

- 1) J.Dongarra, *LINPACK: users' guide*, ser. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1979. [Online]. Available: <http://books.google.co.jp/books?id=AmSm1n3Vw0cC>
- 2) "Top 500 countries share for 11/2011," 2011. [Online]. Available: <http://www.top500.org/charts/list/38/countries>
- 3) "Top 500 press release," 2011. [Online]. Available: <http://www.top500.org/lists/2011/11/press-release>

- 4) “The green500 list november 2011,” 2011. [Online]. Available: <http://www.green500.org/lists/2011/11/top/list.php?from=1&to=100>
- 5) “Top500 performance development,” 2011. [Online]. Available: http://www.top500.org/lists/2011/06/performance_development
- 6) “福島第一原子力発電所,” 2011. [Online]. Available: <http://www.green500.org/lists/2011/11/top/list.php?from=1&to=100>
- 7) “Ati radeon hd 5870 graphics specifications,” 2011. [Online]. Available: <http://www.amd.com/us/products/desktop/graphics/ati-radeon-hd-5000/hd-5870/Pages/ati-radeon-hd-5870-overview.aspx#2>
- 8) “Nvidia tesla c2050 / c2070 gpu コンピューティングプロセッサ,” 2011. [Online]. Available: http://www.nvidia.co.jp/object/product_tesla_C2050_C2070_jp.html
- 9) “Next io vcore extreme - 株式会社 エルザジャパン,” 2011. [Online]. Available: http://www.elsa-jp.co.jp/products/nextio/vcore_extreme/index.html
- 10) G.Chen, L.Chacón, and D.C. Barnes, “An efficient mixed-precision, hybrid CPU-GPU implementation of a fully implicit particle-in-cell algorithm,” *ArXiv e-prints*, Nov. 2011.
- 11) “価格.com eah5870/2dis/1gd5/v2 (pciexp 1gb) 価格比較,” 2011. [Online]. Available: <http://kakaku.com/item/K0000102777/>
- 12) “Nvidia tesla c2070 [pciexp 6gb] 価格比較,” 2011. [Online]. Available: http://kakaku.com/item/K0000264157/?lid=ksearch_kakakuitem.title
- 13) “Ntt-x store,” 2011. [Online]. Available: http://nttxstore.jp/_II_HP13647981
- 14) “Giada gtx580-ddr5 [pciexp 1.5gb],” 2011. [Online]. Available: http://kakaku.com/item/K0000321156/?lid=ksearch_kakakuitem.title
- 15) “Amd display library (adl) sdk,” 2011. [Online]. Available: <http://developer.amd.com/sdks/adlsdk/pages/default.aspx>
- 16) “80 plus verification and testing report,” 2011. [Online]. Available: http://www.acbel.com/ProductFile/80PLUS/ACBEL_PC6024_700W_Report.pdf