Vol.2018-HPC-163 No.21
2018/3/2

Pascal 世代の GPU における電力制約下での性能分析

小野美由紀[†1]　本田巧[†1]　福本尚人[†1]　中島耕太[†1]

概要：本稿では，NVIDIA 社の Pascal 世代の GPU において，電力制約を課して HPC システムの基本的なベンチマークプログラムを実行し，電力制約下での性能を評価した．電力制約として複数の電力キャップおよび動作周波数を設定し，異なる電力制約による性能の違いを分析した．DGEMM や FDTD では平均消費電力が電力制約を下回り，電力制約による影響をほとんど受けなかった．DGEMM のような高負荷処理であっても DGEMM 単体であれば，電力制約下でも，十分に高い性能となることが分かった．メモリバンド幅を測定する STREAM では，最も性能が良いのは最高動作周波数ではなく，1000MHz 未満の低い動作周波数の性能であった．このような現象の原因は明らかになっていないが，性能を最大化するためには，最大動作周波数を設定するのではなく，最適な動作周波数設定が必要であることが分かった．演算がボトルネックとなる HPL では，電力キャップに応じて性能が変化する．電力キャップ下ではある動作周波数で性能が頭打ちとなるため，それ以上の動作周波数にする必要がないことがわかった．

1. はじめに

HPC システムやデータセンターの大規模化と高性能化が進んでおり，これに伴う消費電力の増大が課題となっている．この課題を解決するために，HPC システムにおいて越えてはならない最大の消費電力（以降，電力キャップと呼ぶ）を設定し，この値を超えないように制御することが多い [1]． HPC システムの電力キャップは，気温や供給可能な消費電力に応じて決定される．そのため，同一サーバであっても異なる電力キャップが設定されることが考えられる．ある電力キャップで有効だった性能向上手法がより厳しい電力キャップでは効果がなくなるということも考えられる．

近年，少ない台数で高性能を実現する GPU などのアクセラレータを搭載するシステムが増加している．昨年 11 月に発表された Top500 の上位 5 システムのうち 2 システムに GPU が搭載されている [2]．このような CPU と GPU が混在したシステムでは，電力を制御する対象が増えるため，より複雑になる．

CPU に対する電力制約下での性能に関しては様々な研究が行われている [3]．GPU については消費電力と動作周波数に着目した DVFS (Dynamic Voltage and Frequency Scaling) に関する研究 [4] が行われているが，電力キャップに関する研究はあまり行われていない．さらに，CPU と GPU が混在したシステムにおける電力キャップに関する研究 [5] も行われているが，まだあまり多くはない．

我々は，これまでに CPU に関する電力制約下での性能挙動の分析や最適なアプリケーション実行について検討を行ってきた [6]．さらに，CPU と GPU を統合したシステムにおける電力を考慮した性能最適化技術の確立を目指し，GPU に関する電力制約下での性能特性を調査する．

本稿では HPC システムへの搭載実績のある NVIDIA 社の GPU における電力制約下での性能を測定し，分析を行う．GPU としては，比較的新しく，電力制約と性能の関係に関する分析がまだ十分に行われていない Pascal 世代を用いる．評価プログラムとしては HPC システムの基本的なベンチマークプログラムをいくつか選び，異なる電力キャップや動作周波数における性能，動作周波数，消費電力の分析を行う．分析した結果，メモリバンド幅を測定する STREAM では，最も性能が良いのは最高動作周波数ではなく，1000MHz 未満の低い動作周波数の性能であった．このような現象の原因は明らかになっていないが，性能を最大化するためには，最大動作周波数を設定するのではなく，最適な動作周波数設定が必要であることがわかった．演算がボトルネックとなる HPL では，電力キャップに応じて性能が変化する．電力キャップ下ではある動作周波数で性能が頭打ちとなるため，それ以上の動作周波数にする必要がないことがわかった．

2 章では GPU における電力制約について述べる．3 章では評価に使用するプログラムについて説明する．次に，4 章では性能評価と分析を行い，最後にまとめる．

2. GPU における電力制約

2.1 電力制約下での性能特性

一般的に，アプリケーションの性能チューニングを行うことにより，IPC (Instructions per Cycles) が向上する．これにより，消費電力が増加し，温度も上昇する．アプリケーションの動作環境において，温度の制限や電力の制限が課せられている場合がある．そのような環境において，課せられた制限を超えると，強制的に動作周波数が下げられることにより，性能が低下する場合もある．例えば，性能向上はするが，電力性能が悪くなるような性能チューニングをする場合，アプリケーションの性能チューニングをして

†1 (株)富士通研究所
Fujitsu Laboratories Ltd.

も，効果が得られず，逆に性能が低下することになる．

電力に関する制約は，環境依存であり，状況に応じて異なる制約が課せられる場合がある．電力制約によって性能がどう変化するかを把握していないと，効果的なチューニングが行えない．

2.2 GPU における電力制約

電力制約を課すために，電力キャップを設定する．さらに，その電力キャップ下において動作周波数を指定することにより，より詳細な制約を課して，アプリケーション性能への影響を調査する．複数の電力キャップや動作周波数を設定し，アプリケーションの性能とアプリケーション実行時の消費電力，実際の動作周波数の変化を確認する．また，温度による動作周波数制御などが行われることもあるため，温度も確認する．

GPU には，Fermi 以降のアーキテクチャに対するモニタリングや管理機能を提供する NVIDIA システム管理インタフェースである nvidia-smi (NVSMI) がある．nvidia-smi により電力キャップを設定することができる．また，nvidia-smi ではアプリケーション実行時の GPU 動作周波数とメモリ動作周波数を指定することも可能である．今回はこの機能を利用して電力制約を課す．

また，nvidia-smi には一定間隔毎に指定情報を出力する機能があり，この機能を使用して，電力に関連する消費電力や温度などのデータの採取することが可能である．これにより，電力消費の推移を把握することができ，この出力データを元に統計情報を算出することができる．

一方，関数などのプロファイリングを行うツール nvprof も提供されている [7] ．このツールでは，関数プロファイルだけではなく，指定アプリケーション実行時の GPU やメモリの動作周波数, 消費電力, 温度に関する統計情報（平均値，最小値，最大値）が得られる．

nvidia-smi では出力データを取得できないことがあり，必ずしも完全ではない．しかし，消費電力などのデータの推移を把握することができる．一方，nvprof ではプロファイリングの負荷が生じると予想される．そこで，今回は nvidia-smi によってデータを得ることにする．

3. 評価プログラム

電力特性を評価するプログラムとして，演算がボトルネックとなる倍精度行列積計算 (DGEMM) と High Performance Linpack (HPL)，メモリがボトルネックとなる Finite Difference Time Domain (FDTD) 法と STREAM の GPU プログラムを採用した．

3.1 DGEMM

演算がボトルネックとなる計算として倍精度行列積 (DGEMM) がある．本研究では，NVIDIA が提供する NVIDIA GPU のための数値計算ライブラリである cuBLAS [8] を用いて行列積を行うプログラムを作成した．CPU-GPU 間のデータ転送は最初の一回のみ行う．

3.2 High Performance Linpack

HPL [9] は，Top500 でも使用されるシステムの浮動小数点演算性能を評価するベンチマークプログラムであり, LU 分解を用いて連立一次方程式を解くプログラムである．具体的には，図 3-1 に示すように，LU 分解，行更新，残りの行列に対する行列積の 3 つの処理を残りの行列が無くなるまで繰り返し，これによって得られる行列の LU 分解の結果を用いて，後退代入を行うことで，連立一次方程式を解いている．HPL の計算は倍精度演算であり，処理時間の大半を倍精度行列積が占めているため演算がボトルネックとなる．

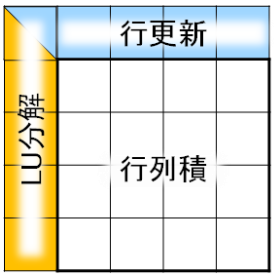


図 3-1　HPL の計算

本研究で用いる HPL プログラムでは, 計算量が最も多い行列積を GPU で行い，残りの処理を主に CPU で行う．高速化のために依存関係の無い処理や CPU-GPU 間のデータ転送は同時に実行している．また, 行列を CPU のメモリ上に配置すると，行列積ごとに発生する CPU と GPU 間のデータ転送量が大きくなり，この転送がボトルネックとなってしまうため，行列データは GPU のオフチップメモリに配置している．

3.3 Finite Difference Time Domain 法

FDTD 法は電磁界の基礎方程式である Maxwell 方程式を差分法による数値計算する方法であり，電磁界関係の多くの分野の研究開発で使用されている．FDTD 法は，計算対象の大規模化・複雑化につれ，多くの計算時間が必要とされる．そのため，HPC 技術を利用して高速化を図り計算時間を短縮することが求められている[10] ．

FDTD 法では，解析を行う空間を格子状に分割し，各格子点の電界と磁界の現在の状態をもとに次の時刻の電界と磁界の状態を計算する．各格子点の電界・磁界の計算はステンシル計算である．特別な最適化を行わない FDTD 法はメモリバンド幅ボトルネックのアプリケーションとしてよく知られている．

本研究で用いる FDTD 法の GPU プログラムは，電界・磁界をそれぞれ計算する 2 つの GPU 関数で構成される．各 GPU 関数では，大量のスレッドを起動し，各スレッドが割り当てられた格子点の電界/磁界計算を行う．本実装では，GPU のオフチップメモリに全てのデータが納まるサイズに問題を限定し，電界・磁界計算時にデータ転送または CPU での計算は行っていない．また，明示的な GPU のオンチップメモリへのキャッシングを行っていない簡単な実装である．

3.4 STREAM

STREAM はメモリバンド幅を計測するベンチマークプログラムである [11] ．大きな 1 次元配列に対して，単純な 4 種類のメモリアクセス処理を行う．元の STREAM ベンチマークプログラムは CPU 向けの実装であるため，CUDA 環境向けにコードが記述されている BabelStream [12] を活用する．この実装では，STREAM のメモリアクセス処理が GPU で実行するように記述されている．

4. 測定と結果の分析

4.1 電力制約下での性能測定

3 章で述べた 4 つの評価プログラムに対して，電力キャップと GPU の動作周波数を変えて実行し，性能を比較する．評価には，CX2570M2 サーバ 1 台を用いた．このサーバには GPU として NVIDIA 社の Tesla P100 が 2 台，CPU として Intel 社の Broadwell が 2 台搭載されている．

測定では，評価プログラムの性能だけでなく同時に GPU の動作周波数，温度，消費電力を計測する．これらの測定には nvidia-smi コマンドを使用する．nvidia-smi コマンドでは，一定間隔毎に GPU の使用率，消費電力，温度などを出力するように引数を指定する．本測定では，プログラム実行直前から実行終了まで，1 秒毎に上記データを出力させる．GPU あるいはメモリが使用されていた場合に出力されたデータの平均値を算出し，これを集計に使用する．GPU およびメモリの使用の有無はそれぞれの使用率から判断する．なお，nvidia-smi コマンドではデータを取得できないこともあり，必ずしも正確な平均値とはなっていない場合もある．しかし，一定時間間隔で値の出力することにより，値の変化を確認することが可能である．

今回評価対象とする GPU では 125W から 250W までの電力キャップを設定できる．デフォルト値は 250W である．今回は 125W から 250W 間隔に 150W, 175W, 200W, 225W, 250W の 6 段階の電力キャップについて評価する．また，GPU の動作周波数としては 544MHz から 1328MHz まで 63 段階設定可能である．デフォルト値は 1189MHz である．性能特性の正確な把握のため，また，HPC プログラムの性能チューニングでは細かな設定も必要となることから，今回は全 63 段階の動作周波数における性能を測定する．なお，メモリ動作周波数は変更できないため，固定である．

表 4-1　評価環境

サーバ	CX2570M2
GPU	Tesla P100 PCIE×2
GPU 動作周波数	544~1328GHz (1189MHz)
電力キャップ	125~250W (250W)
メモリ動作周波数	715MHz
メモリ容量	16GB
メモリバンド幅	732GB/s
L2 キャッシュ	4MB
CPU	Broawell (14 コア)×2

電力制約下における基本的な電力特性を把握するため，HPL は 2GPU での実行，その他は 1GPU (GPU0) での実行のみを対象とする．

電力制約がない状態において，評価プログラムの性能がなるべくよくなるようにパラメータを指定した．また，多数の電力制約での性能測定を行うことを考慮して，実行時間は 30 秒前後になるようにパラメータを指定した．

DGEMM は，行列サイズ A(10240, 384) * B(384, 8960)，メモリサイズ 0.8GB，ループ回数 2000 で実行する．

HPL は問題サイズ N60000，NB384，プロセス数 2，各プロセスのスレッド数 14 とした．

FDTD の問題サイズは 256*256*256，配列サイズは 1152MiB である．

STREAM ベンチマークプログラムは配列サイズが約 0.3GB，合計サイズは 0.8GB，実行回数 5000 回とした．

4.2 性能評価

本節では，各評価プログラムの性能をまとめる．ここで示すグラフは横軸がプログラム実行前に指定したプログラム実行時の GPU 動作周波数，縦軸がプログラム実行により得られた性能である．性能は測定した中での最大値を基準とした相対値で示す．各電力キャップの下，指定動作周波数を変化させて，プログラムを実行し，得られた性能を折れ線グラフで示している．従って，各プログラムの性能グラフには 6 つの折れ線が描かれる．凡例の中の数値は電力キャップを表す. 例えば, ”***-250”は電力キャップ 250W の状況下で，評価プログラムを実行した場合の性能を示している．

図 4-1 は DGEMM の性能を示したものである．性能が最も良いのは 1316MHz や 1328MHz の高動作周波数の場合である．性能は指定動作周波数に応じて向上しており，電力キャップの影響はあまり受けていないことがわかる．ただ

し，電力キャップ 125W の場合には 1290MHz で性能が上がらなくなっている．

図 4-2 は HPL の性能を示している．性能が最も良いのは電力キャップ 250W かつ 1278MHz 以上の高動作周波数の場合であり，電力キャップによる違いが表れている．電力キャップの強さに応じて，ある動作周波数以降の性能が一定になる．

図 4-3 は FDTD の性能グラフである．DGEMM と同様に電力キャップによる違いはあまりない．

図 4-4 は STREAM の性能グラフである．Copy と Mul, Add と Triad の性能傾向は似ている．いずれも最も性能が良いのは最高動作周波数ではなく，1000MHz 未満の低い動作周波数の性能である．なお，最高動作周波数での性能と最も良かった性能との差は 1 から 2%程度である．

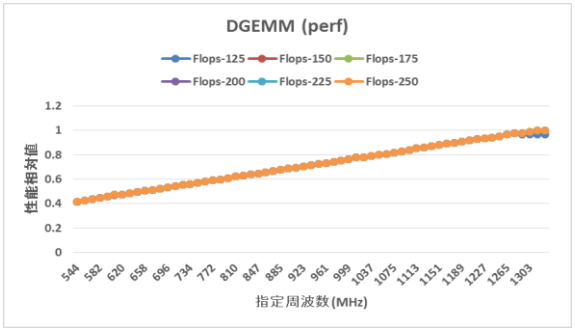


図 4-1　DGEMM の性能

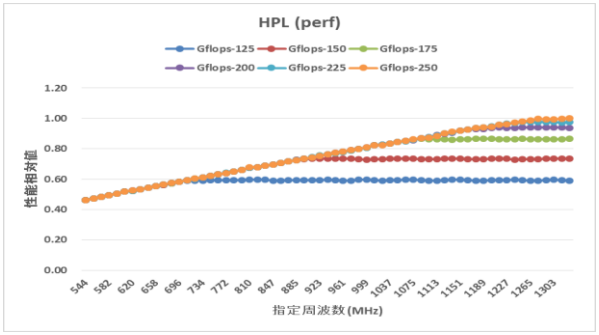


図 4-2　HPL の性能

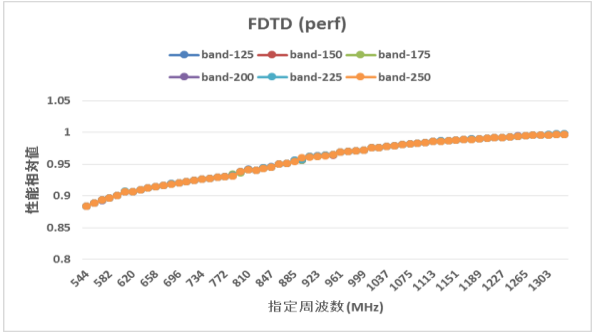
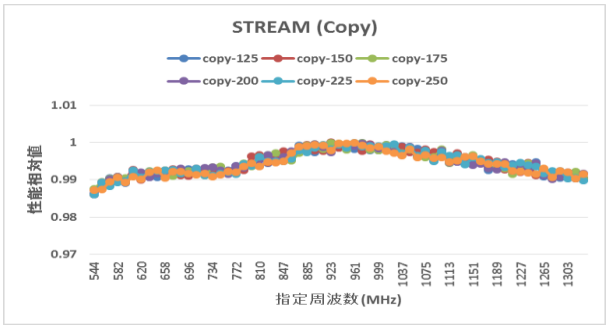
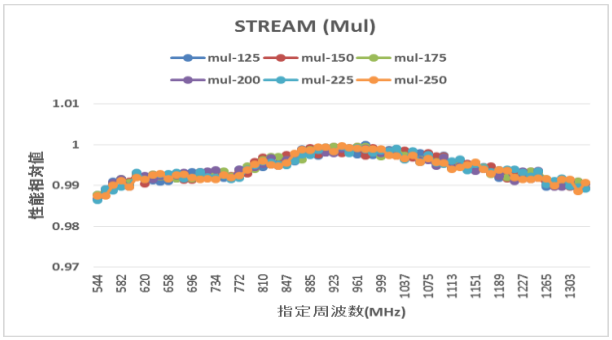
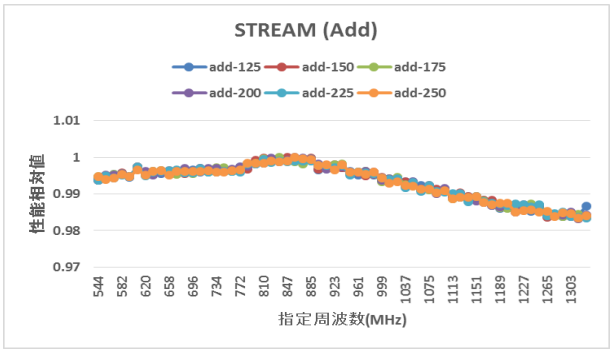


図 4-3　FDTD の性能







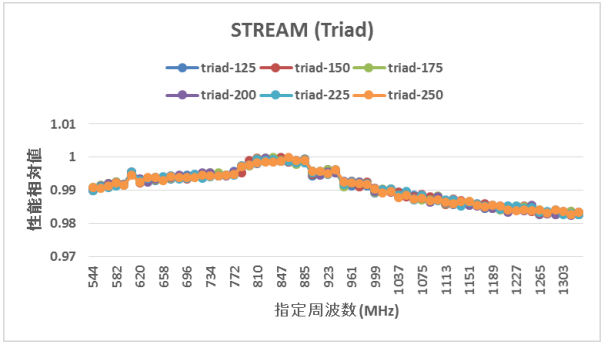


図 4-4　STREAM の性能

4.3　電力制約による性能低下が発生しない事例

DGEMM と FDTD では電力キャップによる性能低下は発生していない．ここでは，FDTD について，分析を行う．図 4-5 は指定動作周波数に対する動作周波数，図 4-6 は指定動作周波数に対する消費電力，図 4-7 は指定動作周波数に対する温度をまとめたものである．各グラフの横軸は指定動作周波数，縦軸は指定動作周波数で実行した際の平均動作周波数，平均消費電力，平均温度である．図 4-5 から動作周波数は指定動作周波数に応じて変化していることがわかる．ほぼ指定した動作周波数で動作できていた．平均

消費電力は最大でも 130W 以下であり，電力キャップ 125W 以外は制限範囲内に収まる．電力キャップ 125W の場合には制限を超えないように抑えられている．

さらに，STREAM では動作周波数と性能に相関が見られず，動作周波数が低い場合の性能のほうが高動作周波数時の性能よりも高いという現象が見られた．この要因については明らかになっていない．

STREAM の動作周波数も FDTD と同様に指定動作周波数と同じであった．また，最高動作周波数と最高性能が得られた動作周波数における消費電力を比較すると，約 20W の差があった．従って，動作周波数を抑えることにより，この分の電力を節約することが可能である．温度に関しては，あまり違いが見られなかった．本測定は 1328MHz から順番に実行しており，1328MHz での実行時には GPU が温まっていなかったかもしれない．

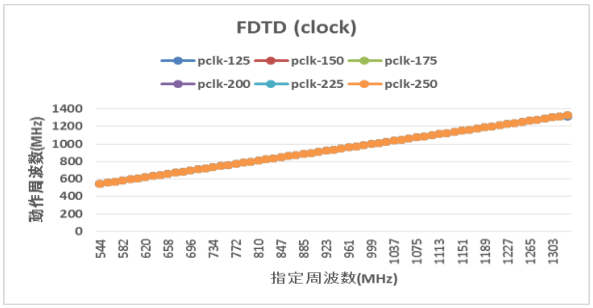


図 4-5　FDTD の動作周波数

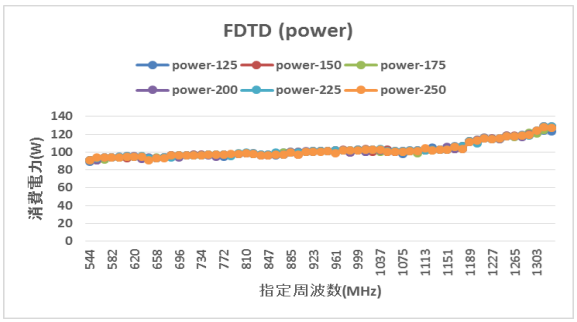


図 4-6　FDTD の消費電力

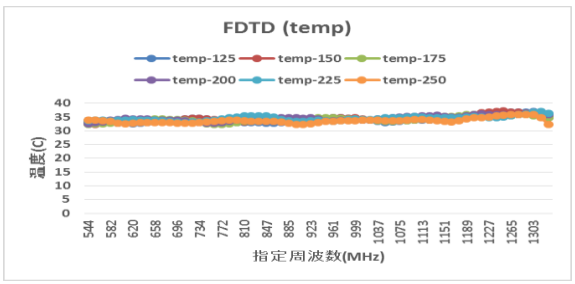


図 4-7　FDTD の温度

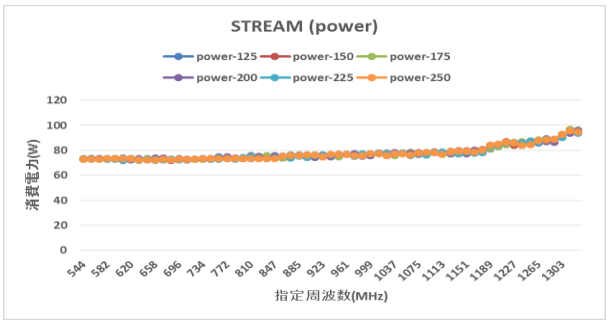


図 4-8　STREAM の消費電力

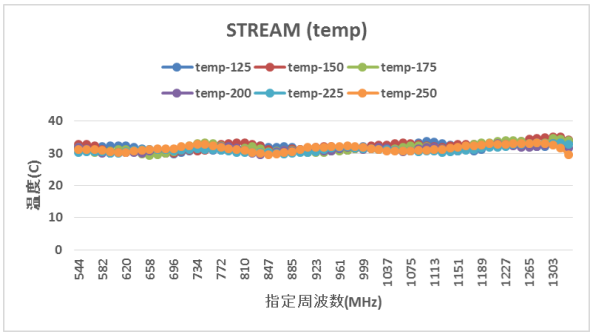


図 4-9　STREAM の温度

4.4　電力制約による性能低下が発生する事例

HPL では電力キャップの強さに応じて性能が変化するが，各電力キャップ下ではある動作周波数以降の性能が一定となった．本測定では 2 つの GPU を使用したが，大まかな傾向は似ているため，ここでは他の評価プログラムでも使用した GPU0 について分析する．図 4-10 に HPL 実行時の指定動作周波数に対する GPU0 の平均動作周波数を示す．動作周波数は，指定動作周波数に応じて増加している．しかし，FDTD や STREAM とは異なり，指定動作周波数よりも低く抑えられている．図 4-11 は電力キャップ 125W，動作周波数 1328MHz の制約下における HPL 実行時の GPU 使用率と動作周波数の変動を表したグラフである．グラフの横軸は経過時間，縦軸左側は動作周波数，右側は GPU 使用率である．GPU 使用率が低いときには動作周波数は設定された 1328MHz になっているが，使用率が高くなると動作周波数が抑えられていることがわかる．今回，GPU 使用率は 0 でない場合の平均を動作周波数としており，このような場合には動作周波数は指定動作周波数を下回ることになる．

図 4-12 は平均消費電力のグラフであり，平均消費電力は設定された電力キャップ以内収まるように抑えられている．平均温度は図 4-13 で示すように電力キャップによる違いはあまり見られないが，電力キャップと動作周波数に応じて若干平均温度は上昇している．

HPL では電力キャップによって性能が変わるが，各電力キャップ下ではある動作周波数で性能が頭打ちとなる．従って，それ以上の動作周波数にする必要はない．

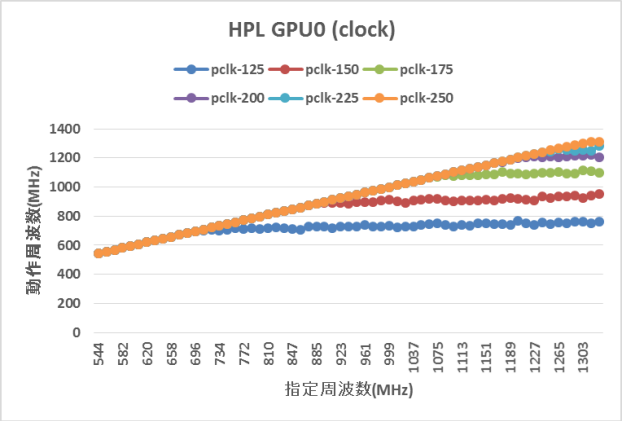


図 4-10　HPL(GPU0)の動作動作周波数

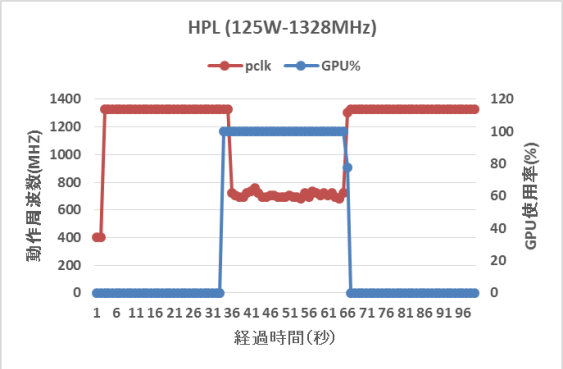


図 4-11　HPL GPU0 の動作周波数変動

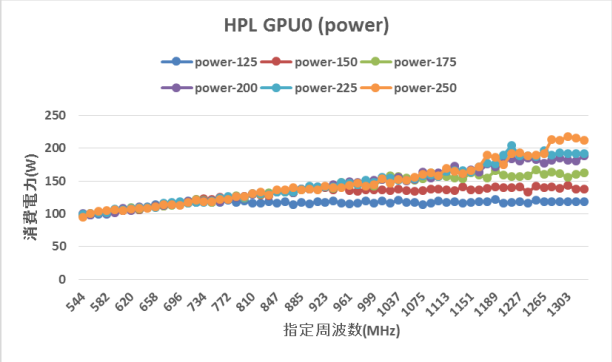


図 4-12　HPL(GPU0)の消費電力

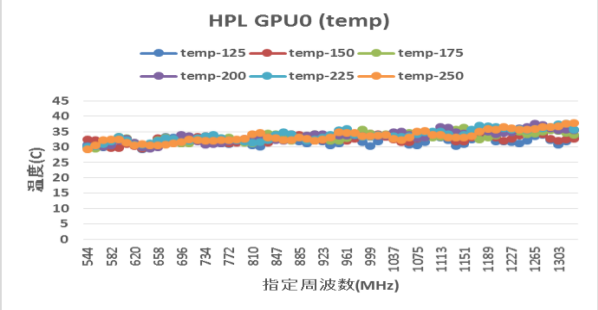


図 4-13　HPL(GPU0)の温度

5.　おわりに

本稿では，NVIDIA 社の Pascal 世代の GPU において，電力制約を課して HPC システムの基本的なベンチマークプログラムを実行し，電力制約下での性能を評価した．電力制約として複数の電力キャップおよび動作周波数を設定し，異なる電力制約による性能の違いを分析した．DGEMM や FDTD では平均消費電力が電力制約を下回り，電力制約による影響をほとんど受けなかった．DGEMM のような高負荷処理であっても DGEMM 単体であれば，電力制約下でも，十分に高い性能となることが分かった．メモリバンド幅を測定する STREAM では，最も性能が良いのは最高動作周波数ではなく，1000MHz 未満の低い動作周波数の性能であった．このような現象の原因は明らかになっていないが，性能を最大化するためには，最大動作周波数を設定するのではなく，最適な動作周波数設定が必要であることがわかった．演算がボトルネックとなる HPL では，電力キャップに応じて性能が変化する．電力キャップ下ではある動作周波数で性能が頭打ちとなるため，それ以上の動作周波数にする必要がないことがわかった．

参考文献

[1] 黄巍 他：エネルギー効率を考慮した電力制約下でのスループット指向ジョブスケジューリング，HPCS2015 (2015)
[2] Top500 : http://www.top500.org/
[3] 稲富雄一 他：電力制約型スーパーコンピュータにおける性能モデリング，HPC-155 No.17 (2016)
[4] Rong Ge et al. (Marquette Univ, USA) : Effects of Dynamic Voltage and Frequency Scaling on a K20 GPU, IEEE Computer Society (2013)
[5] 都筑一希　他　（東工大）, CPU・GPU 混載ノードにおける電力・性能モデルを用いたオンラインパワーキャッピング手法，Vol.2015-HPC-152, No.2
[6] 小野美由紀 他：電力制約下における最適なアプリケーション実行パラメータの導出手法，HPC-161 No.4 (2017)
[7] nvprof : http://docs.nvidia.com/cuda/profiler-users-guide/index.html
[8] cuBLAS: http://docs.nvidia.com/cuda/cublas/index.html
[9] HPL : A. Petitet, R. C. Whaley, J. Dongarra, and A. Cleary. HPL - a portable implementation of the high-performance linpack benchmark for distributed-memory computers. http://www.netlib.org/benchmark/hpl, September 2008.
[10] 深谷猛 他：タイルレベルの並列処理を可能とする地空間タイリング手法を用いた 3 次元 FDTD カーネルの実装と性能評価，HPC-16 No.35 (2017)
[11] McCalpin, John D., 1995: “Memory Bandwidth and Machine Balance in Current High Performance Computers”, IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, December 1995.
[12] BabelStream : https://uob-hpc.github.io/BabelStream/