Vol.2018-HPC-167 No.19
2018/12/18

HPCアプリケーションにおける低精度演算の積極的利用による電力効率改善の検討

坂本 龍一[†1,a)]　近藤 正章[†1]　藤田 航平[†1]　市村 強[†1]　中島 研吾[†1]

概要：近年，数値の精度を落とし低い精度で演算を行う低精度演算が着目されている．低い精度で演算を行うことで小メモリ化・高速化・省エネルギー化を実現することができる．本研究では HPC アプリケーションに対し低精度演算を積極的に利用することによる電力効率の改善について検討する．そこで，ポアソン方程式と地盤震動シミュレーションの前処理に対し，倍精度で計算した場合と単精度で計算した際の電力性能の比較を示す．データの分割方法や並列数等のパラメータが電力性能に対して与える影響を確認する．

1. はじめに

将来の HPC システムでは，消費電力がシステム設計や実行性能を制約する最大の要因となると考えられている．たとえば，本原稿執筆時点で世界最高性能を誇る Summit は 188 ペタフロップスの性能を達成するために 9MW 近い消費電力を要する [1]．しかし，現在の大規模計算機センターの電力設備状況や物理的な制約からするに，将来的にこれ以上の電力供給能力を持つ計算機センターを設置することは難しい．つまり，2020 年ごろに実現されるエクサスケール級のシステムでは，現在と同規模の 10-40MW 程度の電力で，現在の 5 倍近い性能を達成することが必要である．

これらの問題に対し，近年ではハードウェアの技術革新に伴い，様々な電力制御が可能となっている．例えば，DVFS では CPU の動作周波数と動作電圧を変更することによって，性能とのトレードオフとなるが消費電力・消費エネルギーを削減可能である．また，Intel が提供する RAPL[10] は CPU や DRAM が消費する電力をリアルタイムにモニタリングし，CPU が使用する電力を制約する機能を有している．さらにこれらの機能を利用し，省電力化，省エネルギー化を行うための最適化手法の研究 [2][3] が盛んにおこなわれ実際の運用環境において実計算の消費電力の抑制が期待されるようになってきている．

また，Approximate コンピューティング [4] と呼ばれる考え方が近年着目されている．Approximate コンピューティングでは演算やメモリ等へのデータ保存に関し，真の値ではなく，近似的な値を計算・保存するものであり，誤差が生じることを許容し，演算やデータ保存にかかるコストを削減するものである．演算では精度を落とす手法や，IEEE754 に準拠せずに簡単化した浮動小数点演算を用いる研究 [5] がある．メモリでは電源電圧を低下させることで電力を削減する手法等がある．このように信頼性の低下をある程度許容することによって消費電力を大きく削減する．一方で許容できる信頼性の低下率は対象となるアプリケーションによって異なるため，アプリケーションごとの最適化が重要である．例えば画像処理であれば画素における多少の値の違いが人の知覚に与える影響は小さいため，Approximate コンピューティングを導入しやすい．一方で HPC の分野では精度保証が重要であるため積極的な利用は行われていなかったが，近年の電力需要の背景を受け高性能計算分野においても Approximate コンピューティングが着目されつつある．

そこで，本研究では HPC 分野のアプリケーションに対して Approximate コンピューティングを適応することによる効果を確認する．Approximate コンピューティングの手法や戦略は多岐にわたるが，本研究では特に演算精度を落とす低精度演算を積極的に用いることによる計算時間短縮，消費電力の削減について着目する．今回はポアソン方程式のソルバーと地震シミュレーションのソルバーの一部に対し，倍精度を用いて演算を行っていた部分を単精度に変更することによる低精度演算化が消費電力・消費エネルギーに与える効果を確認する．

そこで本研究では，(1) 倍精度を用いていた既存の HPC アプリに対し単精度計算を積極的に導入する方法を示し，(2) 低精度演算を積極的に用いることによる消費電力変化を確認する．合わせて，(3) 演算時間，消費エネルギーの

[†1] 東京大学
[a)] r-sakamoto@hal.ipc.i.u-tokyo.ac.jp

変化について調査する．さらに，(4) 低精度化と合わせて動作周波数変えることによる消費エネルギーの削減について検討を行う．本稿では特に HPC 分野で多く用いられるポアソン方程式と地震シミュレーションのカーネルの一部を低精度化することによる効果について確認する．

2 章では低精度化について示し，3 章では本研究で省電力化を行う HPC アプリケーションについて示す．4 章に評価結果を示し，5 章にまとめを示す．

2. 低精度演算

演算精度を落とした計算として一般の汎用計算機において高精度で計算を行っていた部分を低精度演算で代替えを行ったり，FPGA 等の専用ハードウェアを用いて任意の精度で計算を行う方法がある．汎用計算機を用いた低精度演算の場合，従来 64bit 倍精度演算を用いていた部分を 32bit 単精度を用いて計算する手法や，128bit の 4 倍精度を用いていた部分を 64bit 倍精度で計算するといったものがある．多くの場合，プログラムの修正は軽微であり，容易に修正が可能であるが，データ表現に利用するビット数が半減するため，大きく計算誤差が増加する恐れがある．一方で FPGA を用いた場合，ビット数を任意に変更できるためアプリケーションに必要な精度を保ちつつ低精度化が可能である．一方で高位合成等の技術が普及した 2018 年現在でも演算精度変えつつプログラミングを行うにはシミュレーション時間やデバッグ等に大きなコストが必要である．2018 年現在，HPC 分野では主に高性能な汎用計算機が用いられているため，本研究では汎用計算機を用いたアプローチについて示す．

2.1 消費電力と消費エネルギー

1 回の加減算や乗除算といった演算単位では，低精度化を行ことにより，ビット数の削減により電力が削減されることが期待される．また，メモリからデータを読む際も，必要なデータのビット幅が少なくて済むため，消費電力が下がることが期待されれる．一方でプロセッサ全体で見た場合，SIMD の積極的な利用が行われるため，単精度化する場合，同時に実行される演算数が増加し，演算密度が高くなり，消費電力が増加する可能性が生じる．例えば，Intel AVX[6] の場合，64bit 倍精度演算は同時に 4 並列で演算が可能だが，32bit 単精度を用いる場合 8 並列での演算が可能となる．低精度化によって並列数が増加することでスループットは倍増するが，演算密度が上昇するため電力が上昇する可能性がある．一方でスループットが 2 倍になるため実行時間の短縮により消費エネルギーの削減が期待できる．

2.2 精度保証

低精度化を行うことで計算誤差が増加し，有効な演算結果が得られない恐れがある．そのため，低精度化を行う際は必要な精度が満たされているかについて注意深く確認することが重要である．また，反復計算を行うような処理の場合，低精度化により残差誤差の収束性が悪化することが考えられる．そのため，倍精度と単精度を混合して利用する混合精度を用いる方法も重要である．本研究で評価を行うポアソン方程式ソルバーについてはプログラム全体を単精度化することとし，地震シミュレーションについては，一部のカーネルのみを単精度化する混合精度演算を用いることとした．今回は Fortran のプログラムを直接修正することで低精度化を行ったが，所望の精度を保ちつつ最適な精度を自動的にチューニングする方法は今後の課題である．

3. 低精度演算化するアプリケーション

本研究では HPC システムにおける積極的な低精度演算の初期評価として HPC 分野で多く利用されるポアソン方程式ソルバーと地震シミュレーションアプリに対し低精度化を行い，低精度化の効果を確認する．

3.1 ポアソン方程式 (ICCG 法)

有限体積法によるポアソン方程式から導かれる対称正定な疎行列を計数とする連立一次方程式を解くための不完全コレスキー分解前処理付き共役勾配法 (Preconditioned Conjugate Gradient Method by Incomplete Cholesky Factorization, ICCG 法) を対象に，低精度演算を利用することによる効果を確認する．特に今回は ICCG 法をマルチスレッドを用いて並列化を行ったコード [7] を対象にする．ICCG 法では依存を持たない要素群を同じ色に色付けすることによって，色内での並列処理が可能となる．色数を増やすことによって並列性とスレッド間の負荷分散を実現できる．色付けを行うことで並列処理が可能となるが，同じスレッド内に属する要素は連続の番号でないため，メモリアクセスの際に効率が低下する．このような要素の配置を Coalesced Numbering と呼ぶ．さらに，Coalesced Numbering に対し，同一スレッド内で処理するデータを連続に再配置するように並び替えた Sequential Reordering と呼ばれる最適化が可能である．また，本評価では差分格子のメッシュサイズは 128 x 128 x 128 の直方体とした．単精度化するにあたり，収束判定後の残差誤差について許容内であることを確認した．

3.2 地震シミュレーション

地表の構造物や地盤の堆積・浸食等を考慮した複雑な地盤振動シミュレーションには大規模 Unstructured FEM 解析が適している．しかしながら，大規模 Unstrctured FEM 解析には計算コストが大きいため高速化が重要である．本評価では市村 [8][9] らが提案する様々な高性能化手法を融合させて地盤振動問題を解く Adaptive CG ソルバーに着

表 1　Reedbush-L のノード構成

プロセッサ名	Intel Xeon E5-2695v4 (Broadwell-EP)
プロセッサ数 (コア数)	2 (36)
周波数	2.1 GHz (Turbo boost 時最大 3.3 GHz)
理論演算性能	1209.6 GFlops
メモリ容量	256 GB
メモリ帯域幅	153.6 GB/sec

目する．Adaptive CG は線形方程式の反復法ソルバーの一種である共役勾配法 (CG 法) の前処理において，固定された行列 M を使って探索方向 z ← Mr を定めるのではなく，対象とする方程式 Ax=f の A による前処理方程式 r=Az を解いた z を前処理の探索方向に使うアルゴリズムである．本手法では，前処理行列は荒く解くことができる．そのため，この前処理部分に単精度を用いることが可能である．本地震シミュレーションの評価ではこの前処理部分に着目し，倍精度を用いた場合と単精度を用いた場合との比較を示す．

4. 評価

上記で述べた 2 つのアプリケーションに対する電力・エネルギー評価を示す．評価では倍精度を単精度化することによる消費電力の変化，実行時間の変化，消費エネルギーの変化を確認する．

4.1 評価環境

評価には東京大学情報基盤センターの Reedbush-L スーパーコンピュータシステムを用いた．計算ノードの仕様を表 1 に示す．また，本評価では低精度演算を利用する初期評価であるため，1 台の計算ノードのみを利用するシングルノードジョブとして評価を行う．また，製造プロセスのバラつきにより実行するノードによって電力評価結果が異なる場合があるため，本評価ではノード固定の設定を与え，すべての評価が同一の計算ノードで実行されるようにした．

電力計測方法

電力評価には Intel が提供する CPU の電力制御機能の RAPL[10] を用いる．RAPL は SandyBridge 以降の Xeon プロセッサで利用できる機能であり，CPU パッケージや DRAM の消費電力や消費エネルギーを計測することが可能である．更にプロジェクト内で作成した電力計測ライブラリを導入し消費電力，消費エネルギーの計測を行う．本電力計測ライブラリを用いることで，プログラム中の任意の区間の計測を行うことができる．ポアソン方程式ソルバーでは ICCG 法の関数部分の計測を行い，地震シミュレーションについては前処理部分のみを計測の対象とした．また，Reedbush システムは 2 ソケットのシステムであるが，1 つ目のソケットのみをプログラムが利用するように OpenMP の設定を行い，1 ソケット目のみの電力を

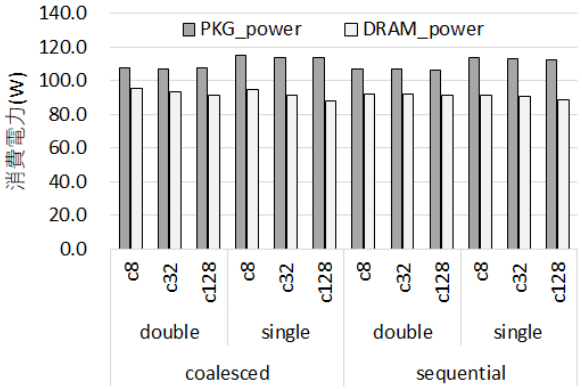

図 1　倍精度 (double) を単精度 (single) 化した場合の ICCG 関数の平均消費電力の比較

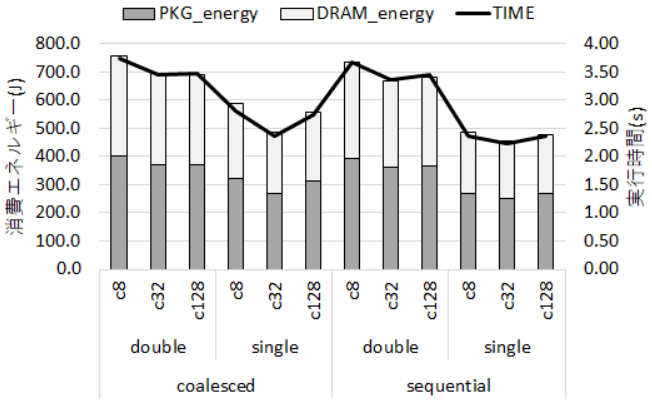

図 2　倍精度 (double) を単精度 (single) 化した場合の消費エネルギーの比較

計測するようにした．また，プログラムのコンパイルには Intel compiler 2018 を用い，-O3 オプションを用いた．

4.2 評価結果

ICCG 法

本評価では低精度演算利用時における電力を確認するため，精度を変えて (double, single) 計算を行った場合の平均消費電力を評価した．また，データ配置方式 (coalesced, sequential)，分割数 (c8, c32, c128) を変えて評価を行った．図 1 に消費電力の結果を示す．図中の PKG power は CPU の消費電力を示しており，コアやキャッシュが消費する電力が含まれている．また，DRAM power は 1 つのソケットに接続されたすべての DIMM が消費する電力を示している．double と single を比較すると倍精度を単精度化することにより CPU 電力 (PKG power) が増加したことがわかる．これは，演算密度が上昇したためである．パフォーマンスカウンタの値を確認したところ IPC も 2 倍程度に上昇していた．coalesced と sequential を比較した場合，sequential の方が CPU の消費電力が 1〜2W 程度低いことが分かっ

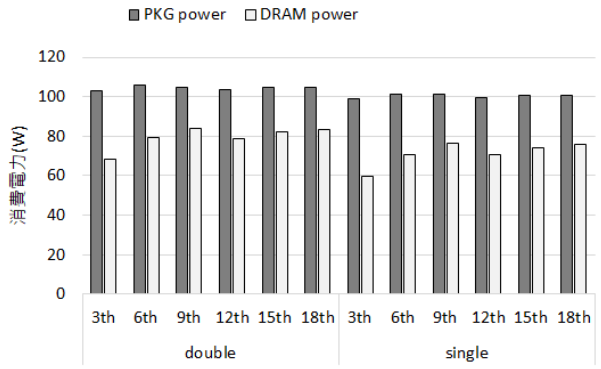

図 3 倍精度 (double) を単精度 (single) 化した場合の Adaptive CG の平均消費電力の比較

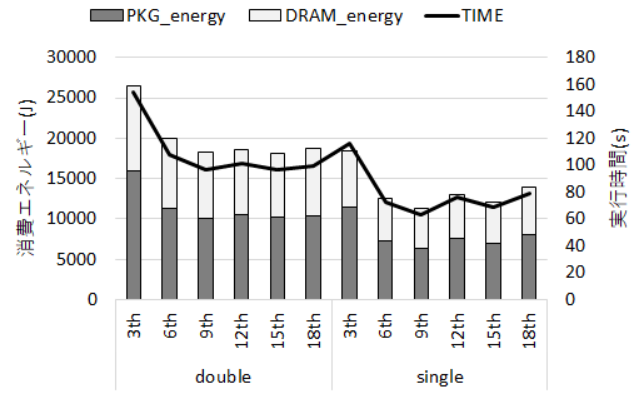

図 4 倍精度 (double) を単精度 (single) 化した場合の消費エネルギーの比較

た．また，DRAM の電力も同じく 1〜2W 程度 sequential の方が低い結果となった．また，データの分割数を 8, 32, 128 と増やすことによって DRAM の電力が低下することが分かった．データ分割数を増やすと CPU キャッシュの利用効率が低下したためだと考えられる．次に実行時間と消費エネルギーに関して評価を行った．この結果を図 2 に示す．低精度化することにより実行時間が大きく短縮されることが確認でき，これに伴い消費エネルギーも大きく削減された．特に，sequential で色数が c8 の場合，倍精度を単精度化することによって最大で 34%近いエネルギー削減を実現した．また，coalesced より sequential の方が削減率が大きい．coalesced の場合，倍精度を単精度化することによって，平均 23.7%エネルギーを削減できるが，sequential の場合，平均 32.1%のエネルギーを削減することができた．分割数を増やしすぎると実行時間が長くなり，エネルギー効率が悪化した．

地震シミュレーション

Adaptive CG の前処理部分の電力を確認するため，精度 (double, single) を変えて計算を行った場合の平均消費電力を評価した．また，計算に用いる OpenMP のスレッド数を 3 スレッド，6 スレッド，9 スレッド，12 スレッド，15 スレッド，18 スレッドと変化させた．平均消費電力の結果を図 3 に示す．これらの結果，ICCG 法とは対象的に倍精度 (double) を単精度化 (single) 化することにより，CPU 電力 (PKG power) と DRAM 電力 (DRAM power) がともに低下することが確認された．倍精度の CPU 電力の平均は 104.4W であったが，単精度の CPU 電力の平均は倍精度よりも 4W 程低い 100.3W であった．また，倍精度の DRAM 電力の平均は 79.4W であったが，単精度の場合，8W 程度低い 71.3W であった．詳細は確認中であるが，Adaptive CG は非常に大きなデータを扱うため，LLC ミス率が 50%近くと非常に高いことが関係していると考えられる．次に実行時間と消費エネルギーの評価結果を図 4 に示す．この結果，倍精度の場合も単精度の場合も OpenMP のスレッド数を 9 にしたほうが実行時間が短くなることがわかった．評価に用いた Reedbush の CPU はソケット当たり 18 コアの物理コアを有しているが，半数のコアを用いたほうが実行時間が短くなる結果となった．また，倍精度と単精度を比較した場合，ICCG 法と同様に単精度化することにより実行時間が大幅に短くなることが確認できた．その結果，単精度で 9 スレッドを用いた際に最もエネルギー効率が良くなることがわかった．倍精度で 9 スレッドを用いていたものを単精度にすることで，最大で 38.3%のエネルギーを削減することができた．また，平均で 32.4%のエネルギー削減となった．

5. 動作周波数を変えた場合のエネルギー消費

上記の評価では動作周波数について特に制約を与えない状態でプログラムを実行していたため，常にすべてのコアが最大の周波数で動作していた．そこで，CPU 動作周波数を下げた場合の評価を行う．メモリインテンシブなアプリケーションの場合，メモリアクセスに性能が律速されるため CPU の動作周波数を下げても実行時間は大きく低下しないことが知られている．一方，CPU の消費電力を削減でき，消費エネルギーの削減が期待される．Reedbush ではプロセッサの動作周波数の変更ができなかったため，本評価は Reedbush ではなく著者の研究室にある Intel Haswell CPU を搭載したマシンで評価を行った．また，ICCG 法のみに着目し評価を行う．本評価で用いた Haswell マシンの仕様を表 2 に示す．また，プログラムのコンパイルには GFortran を用い，-O3 オプションを指定した．

低い周波数で演算を行った場合の消費電力と消費エネルギー

初めに，評価に用いるプロセッサがサポートする最低の周波数 (1.2Ghz) を設定した場合と最大の周波数 (2.7Ghz) を用いた場合の消費電力について確認を行った．ICCG 法の色数については最も実行時間が短くなる 32 を選び，周

表 2　Haswell マシンのノード構成

プロセッサ名	Intel Xeon E5-2690 v3 (Haswell)
プロセッサ数 (コア数)	2 (24)
周波数	2.6 GHz (Turbo boost 時最大 3.5 GHz)
理論演算性能	1209.6 GFlops
メモリ容量	256 GB
メモリ帯域幅	136 GB/sec

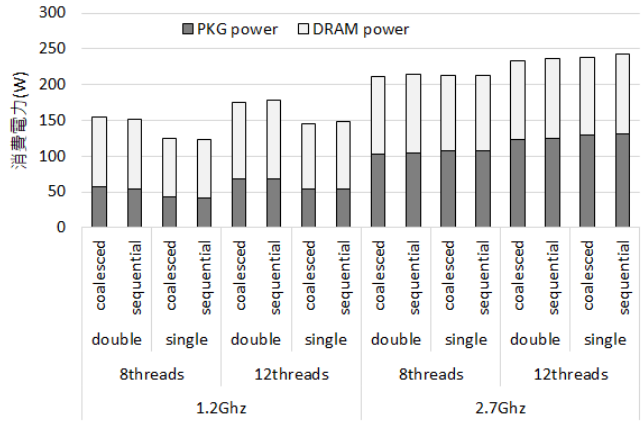

図 5　低い周波数で ICCG 関数を実行した場合の消費電力

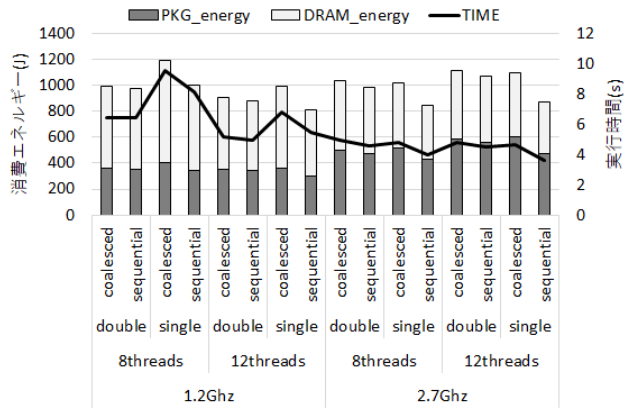

図 6　低い周波数で ICCG 関数を実行した場合の消費エネルギー

波数とスレッド数をパラメータとした．プロセッサの物理コア数は 12 であるため，OpenMP のスレッド数を 8 にした場合と，12 にした場合での比較を行った．これらの消費電力の計測結果を図 5 に示す．これらの結果より，周波数を下げると CPU の電力が大きく低下するが，DRAM 電力の下げ幅よりも CPU 電力の下げ幅が大きいことがわかる．また，周波数が低い場合 (1.2Ghz) に低精度の方がより電力が低くなる．また，OpenMP のスレッド数は 8 スレッドを用いた方が，12 スレッドよりも消費電力が低いことが分かった．

次に，実行時間と消費エネルギーについての結果を図 6 に示す．この結果，周波数を落とすことによって実行時間が延びることが分かった．また，周波数を最大とした場合は 12 スレッドよりも 8 スレッドの方がエネルギー効率が

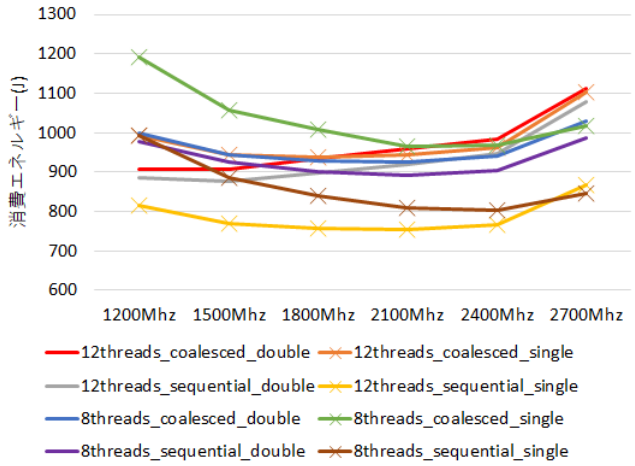

図 7　プロセッサの動作周波数と消費エネルギー

良いことが分かった．また，消費エネルギーを最小化する設定は 1.2Ghz, 12threads, single, sequential であり，単精度を用いて周波数を落とし，スレッド数を増やしたほうが良いことが分かった．

最後に周波数を連続的に変更した場合の消費エネルギーについての評価を行った．本評価では CPU の周波数を 1.2Ghz, 1.5Ghz, 1.8Ghz, 2.1Ghz, 2.4Ghz, 2.7Ghz と変化させた．また，今回は CPU と DRAM のエネルギーの合計を示している．これらの結果を 7 に示す．これらの結果より，周波数が低い場合，実行時間が延びるためエネルギーが増加し，逆に周波数が高い場合，消費電力が増加するためエネルギーが増加する結果となった．また，エネルギーを最小化する周波数はパラメタによって異なる結果となった．スレッド数が多い場合，周波数は低いほうが良い．また，単精度を用いる場合，2.4Ghz が良い結果となった．倍精度で 12 スレッドを利用する場合，1.5Ghz エネルギー効率が良いが，倍精度で 8 スレッドを用いる場合，2.1Ghz が良い結果となった．

6. 関連研究

ここでは，入力に用いるデータや中間データの精度（ビット幅）を変更する Approximate コンピューティングの関連研究を示す．Yeh らは物理シミュレーションにおいて動的な精度変更を行うことによって演算効率を向上させる方法 [11] を提案している．提案手法ではアプリケーションの設計段階においてプロファイルを行い，必要最低限の精度を探索する．また，ランライムに不安定な状態になっていないかを確認する手法を提案している．さらに，これらの結果を基に階層的な FPU を持つアーキテクチャの資源利用を最適化する．アプリケーションに必要な精度に応じて FPU を使い分ける手法を提案している．

Tian らはクラスタリング問題を解く際にノードの距離が十分に離れている場合は荒い粒度でもエラーが生じないこ

とに着目し，off-chip メモリアクセスを行う際のビット幅を削減することで off-chip メモリアクセスを抑制し省エネルギー化を行う手法 [12] を提案している．提案手法では各イタレーションごとにエラーを確認し，適切なメモリビット幅を選択するようにしている．

7. まとめ

本論文では HPC アプリケーションに対し低精度演算を積極的に用いることによる効果について示した．2 つのアプリケーションに対し低精度化を行った結果，LLC ミスが低いアプリの場合演算密度の上昇により CPU の消費電力が若干上昇することが分かった．一方で，LLC ミスが非常に高いアプリは単精度化を行った場合，消費電力が低下するものと考えられる．しかしながら，どちらの場合においても単精度化によって演算スループットが大きく向上し実行時間が短縮するため，消費エネルギーを大幅に削減することができることが分かった．

今後は他の様々なアプリケーションに同様の計測を行い，より詳細な分析を行う．さらに，計算中に精度を切り替えることによるエネルギー削減を検討する方法を検討する．

謝辞 本研究は，学際大規模情報基盤共同利用・共同研究拠点の支援による (課題番号: jh180023-NAH)．

参考文献

[1] TOP500 Supercomputer Sites `https://www.top500.org/`
[2] Y. Inadomi, et al. : Analyzing and Mitigating the Impact of Manufacturing Variability in Power-constrained Supercomputing, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC). 2016.
[3] O. Sarood, et al. : Maximizing throughput of overprovisioned HPC data centers under a strict power budget, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC). 2014.
[4] S. Mittal : A Survey of Techniques for Approximate Computing, ACM CSUR 48-4, 2016.
[5] D. Harris : An exponentiation unit for an OpenGL lighting engine, IEEE Transactions on Computers (Volume: 53 , Issue: 3 , March 2004)
[6] `https://www.intel.co.jp/content/www/jp/ja/architecture-and-technology/avx-512-overview.html`
[7] 中島研吾 : 前処理付きマルチスレッド並列疎行列ソルバー， IPSJ 第 139 回 HPC 研究会，2013 年
[8] T. Ichimura, et al. : Physics-Based Urban Earthquake Simulation Enhanced by 10.7 BlnDOF × 30 K Time-Step Unstructured FE Non-Linear Seismic Wave Simulation, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC). 2014.
[9] T. Ichimura, et al. : A Fast Scalable Implicit Solver with Concentrated Computation for Nonlinear Time-Evolution Problems on Low-Order Unstructured Finite Elements, IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2018.
[10] `https://01.org/rapl-power-meter`
[11] Thomas Y. Yeh, et al. : The art of deception: Adaptive precision reduction for area efficient physics acceleration, In International Symposium on Microarchitecture. 2007.
[12] Ye Tian, et al. : ApproxMA: Approximate memory access for dynamic precision scaling, In ACM Great Lakes Symposium on VLSI, 2015.