

科学研究費助成事業（学術研究助成基金助成金）研究成果報告書

平成25年6月11日現在

機関番号：32613

研究種目：挑戦的萌芽研究

研究期間：2013～2013

課題番号：24650014

研究課題名（和文）

GPUコアを内蔵するプロセッサによる代数的マルチグリッド法の高速化

研究課題名（英文）

Acceleration of algebraic multigrid solver on APU

研究代表者

藤井昭宏 (FUJII AKIHIRO)

工学院大学・情報学部・講師

研究者番号：10383986

研究成果の概要（和文）：

CPU コアと GPU コアが密に結合されている APU 上で、代数的多重格子法の効率的な実装手法の研究を行った。CPU-GPU 間の通信の最小化と、GPU コアは倍精度に対応していないこと、計算性能の効率的利用の観点から、GPU コアで単精度で反復し、CPU コアの上で倍精度に補正する混合精度代数的多重格子法を実装、評価した。その結果、代数的多重格子法において、混合精度手法は有効に機能し、典型的なポアソン問題で収束条件が相対残差の2ノルムで $1.0E-7$ のときに、精度の補正は最大で3回程度で、CPU-GPU 間の通信はほとんど発生しないことがわかった。本手法は Ivy Bridge や Haswell にも適用可能であり、今後分析を進めて行く予定である。

研究成果の概要（英文）：

We studied efficient implementation technique of algebraic multigrid method on APU which has CPU cores and GPU cores in one chip. In order to minimize the communication between CPU cores and GPU cores, and to exploit GPU cores' performance at most, we tested the mixed precision solver that iterates algebraic multigrid method with single precision on GPU cores, and calculate the residual vector with double precision on CPU cores. As a result, mixed precision method works well for algebraic multigrid method. For typical Poisson problems with convergence criterion of relative residual norm of $1.0E-7$, precision correction needs only 3 times in our numerical test and there are little CPU-GPU communication. This method can be applied to Ivy Bridge and Haswell.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
交付決定額	3,000,000	900,000	3,900,000

研究分野：総合領域

科研費の分科・細目：情報学・ソフトウェア

キーワード：代数的多重格子法，混合精度解法，GPGPU

1. 研究開始当初の背景

GPU をアクセラレータとして活用する研究が盛んに行われ、CPU だけ利用する計算よりも10倍以上高速に計算できる例が多く報告されてきた。ただ、アプリケーションのすべての部分が GPU 上で動作することは少なく、多くの場合は CPU で処理する部分と GPU で処理する部分があり、そのような場合、CPU と GPU 間の通信時間が計算性能に大きく影響を

与えることが複数報告されている。CPU コアと GPU コアが密に結合されているシステムが近年利用できるようになり、今後、重要な計算環境の一つになると考えられている。このような環境では新しい最適化技術が重要になる可能性もある。そこで、高速線形解法として注目されている代数的多重格子法を研究対象とし、上記のようなシステムでのアルゴリズム上の工夫も含めた実装手法の研

究を行った。

2. 研究の目的

CPU コアと GPU コアが 1 チップで実装されている APU を利用し、その上での代数的多重格子法の実装手法の検討を行い、プロトタイプ実装による評価を行う。

3. 研究の方法

APU では CPU-GPU 間の転送が高速にできることや、GPU コアで高速に並列処理が可能であること、GPU コアでは特に演算精度によって性能が大きく変化する特徴がある。また、対象としている代数的多重格子法は自由度が大きい線形問題と小さい線形問題に交互に緩和法を適用し、解を効率よく収束させる手法である。

本研究では上記の観点から以下の二つの手法を考察した。

(1) 自由度の数が大きい線形問題は GPU コアで処理し、小さい線形問題は CPU コアで処理する手法

(2) GPU コアでは単精度で演算を行い、適切に CPU コアで倍精度に精度補正する混合精度手法

(1) と (2) とともに CPU と GPU の間に通信が発生する手法であるが、(1) では少なくとも 1 反復あたり 1 回は CPU-GPU 間の通信が必要になると同時に、通常は各サイズの問題ごとに順番に緩和法を適用する必要があり、小さいサイズの問題を処理する時間の割合を考えると (1) の有効性は低い。但し、超高並列環境、もしくはすべての階層を同時に処理するなど多重格子法のアルゴリズムを変更する場合も含めると、より詳細な評価が必要である。一方、(2) では CPU コアと GPU コア間の通信回数にもよるが、GPU の計算性能は有効に利用できる。

本研究では 1 個 APU 上での最適化を考えているため (2) を中心に、有効性を検証する。

代数的多重格子法の反復解法部を OpenCL で実装し評価を行った。

こちらで実装し評価した混合精度手法は下記の 5 ステップである。与えられる線形問題を $A_d x_d = b_d$ とし、添え字が精度を表す。d が倍精度、f が単精度を意味する。

1. $x_d = 0$
2. $r_f = b_d - A_d x_d$
3. if (convergence) stop
4. solve $A_f x_f = r_f$ on GPU core
5. $x_d = x_d + x_f$ goto 1.

上記の 1, 2, 3, 5 が CPU コアで処理され、4 のみ GPU コアにより単精度にて代数的多重格子法で解く。4 は単精度で収束する範囲で収

束条件を設定し、5, 2 により解の精度が倍精度に補正される。

精度が補正されるごとに CPU との通信が起きるが、代数的多重格子法でこの通信が発生する頻度を調べる。また、APU での評価だけでなく、CPU 上でも評価し混合精度手法の反復回数の増加のデメリットとメモリアクセスのメリットについて分析する。

4. 研究成果

Trinity 世代の APU (A10-5800K) と GTX680 を使い、OpenCL を用いて等方性、異方性の 3 次元ポアソン問題に対して評価を行ったところ、結果が図 1 のようになった。

“mul_iso_40_40_40” は等方性での問題サイズが 40x40x40 の場合をさし、ani とついているものは異方性問題を示す。また横軸の各ラベルは実行環境を表し、“CPU” のみ通常の倍精度で計算を進めているが、それ以外は混合精度で処理している。“CPU MIX” は CPU で混合精度で 4 スレッドで計算したものの、“APU/READ_WRITE, ALLOC_HOST” は APU 上で CPU との通信するためのデータの指定方法を READ_WRITE や ALLOC_HOST に指定したものの、“GPU” は GTX680 上で同一 OpenCL コードを動作させたものを指す。縦軸は収束時間を示し、“level0-4” が各レベルでの処理時間、“inter level op” はレベル間移動の時間、“norm” と書かれているのは倍精度版の残差計算の時間、residual_norm_d, f は混合精度演算における倍精度、単精度での残差計算の時間を表す。表 1, 2 には問題サイズが一番大きかったときの収束までの精度ごとの反復回数を記載している。“APU1”、“APU2” はそれぞれ “APU/READ_WRITE”、“APU/ALLOC_HOST” に対応している。

まず表 1, 2 から混合精度計算において、精度補正のループがほとんど行われなことがわかる。等方性、異方性の問題ともに、3 回しか行われなかった。結局、ほとんどの処理は単精度で行えていることになる。次に CPU 上で混合精度演算を倍精度演算と比較すると、単精度での収束計算を繰り返し適用するため、反復回数が増加し収束時間として 1 割程度性能が劣ることが多かった。ただ、反復回数の増加が少なかった場合は、メモリアクセス量が減るため、逆に 1 割ほど高速になる場合もあった。GPU コアを利用した混合精度演算は、APU 上の CPU コア単体での利用に比べると、2 倍以上高速になった。混合精度解法は CPU との通信回数も少ないため、より性能の高い GPU を利用した場合、APU よりもさらに倍程度の性能が出た。APU では通信の最適化のため複数のデータ型を試したが、通信はほとんど発生しなかったため、その違いによる性能の差はほとんどなかった。

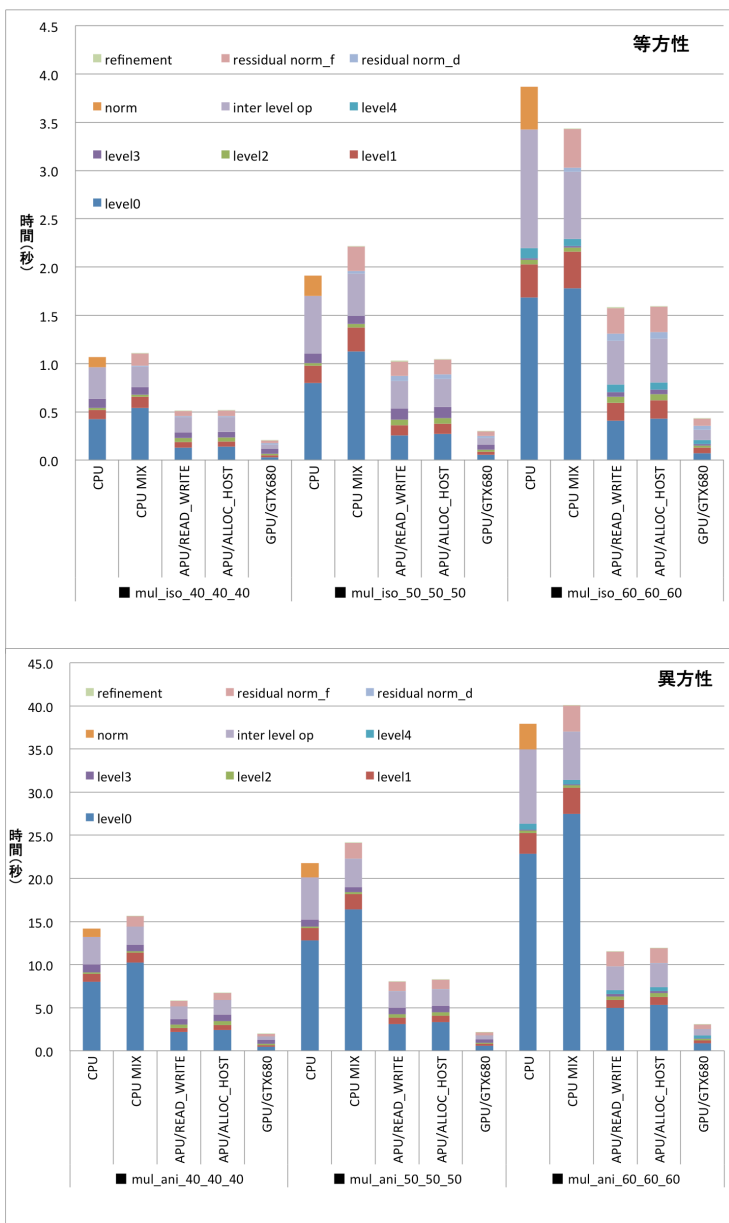


図 1 等方性，異方性問題での収束時間

表 1 等方性問題の反復回数

問題サイズ(60x60x60)

	CPU	CPU MIX	APU1	APU2	GPU
単倍		3 2	3 8	3 8	3 7
倍	3 1	3	3	3	3

表 2 異方性問題の反復回数

問題サイズ(60x60x60)

	CPU	CPU MIX	APU1	APU2	GPU
単倍		2 4 6	2 4 6	2 4 6	2 4 6
倍	2 1 1	3	3	3	3

結果として，混合精度代数的多重格子法は GPU 単体で計算する場合，もしくは CPU による精度補正を行う場合，ともに非常に有効な実装手法となることがわかった。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 4 件)

(1) 高橋光佑, 藤井昭宏, 田中輝雄, マルチカラー法を用いたマルチ GPU 上での AMG 法, 電子情報通信学会論文誌 D, 査読有, Vol. J96-D No. 3, 2013. pp. 452-460.

(2) Akira Nukada, Kento Sato and Satoshi Matsuoka, Scalable Multi-GPU 3-D FFT for TSUBAME 2.0 Supercomputer, Proc. of 2012 ACM/IEEE International Conference for High Performance, Networking, Storage, and Analysis (SC' 12). pp. 1-10.

(3) 高橋光佑, 藤井昭宏, 田中輝雄, GPU のダイレクト通信を用いた AMG 法, 情報処理学会 ハイパフォーマンスコンピューティング研究会研究報告, 査読無, Vol. 2012-HPC-137 No. 30, 2012. pp. 1-8.

(4) 荒谷祐紀, 藤井昭宏, 田中輝雄, GPU 上での AMG 法における Chebyshev 多項式緩和法, 情報処理学会 ハイパフォーマンスコンピューティング研究会研究報告, 査読無, Vol. 2012-HPC-137, No. 36, 2012. pp. 1-7.

[学会発表] (計 6 件)

① Akira Nukada, Performance of 3-D FFT using Multiple GPUs with CUDA 4, NVIDIA GPU Technology Conference 2012, 2012 年 05 月 14 日~2012 年 05 月 17 日.

② 額田 彰, CUDA 版自動チューニング手法, GPU Technology Conference Japan 2012, 2012 年 07 月 26 日~2012 年 07 月 26 日.

③ Akihiro Fujii, Teruo Tanaka, Online Auto-Tuning Technique for Algebraic Multi-Grid Solver, Conference on Advanced Topics and Auto Tuning in High Performance Scientific Computing, 2013 年 03 月 27 日~2013 年 03 月 29 日.

④ Kosuke Takahashi, Akihiro Fujii, Teruo Tanaka, Modified Multi Color Based AMG Method on Multiple GPUs, The 11th International Symposium on Advanced Technology, 2013 年 10 月 30 日~2013 年 10

月 30 日.

⑤ Akihiro Fujii, Online Auto-tuning Technique of AMG Solver for Fluid Analysis, The 11th International Symposium on Advanced Technology, 2013 年 10 月 30 日～2013 年 10 月 30 日.

⑥ 藤井昭宏, 中村修, 流体解析を対象とした AMG ライブラリの自動チューニング技術, 計算工学講演会, 2013 年 05 月 30 日～2013 年 05 月 30 日.

6. 研究組織

(1) 研究代表者

藤井 昭宏 (FUJII AKIHIRO)
工学院大学・情報学部・講師
研究者番号 : 10383986

(2) 研究分担者

額田 彰 (NUKADA AKIRA)
東京工業大学・学術国際情報センター・産学連携研究員
研究者番号 : 40545688