

jh150046-NA24

超大規模シミュレーションのためのアーキテクチャ 特性を考慮した通信削減技術

遠藤敏夫（東京工業大学）

概要 メモリの速度性能・容量の伸びが、メニーコア化するプロセッサの伸びに追いつかないというメモリウォール問題は、今後のスパコンアーキテクチャにおいて顕著となると見られている。この問題により、今後も高性能化・大規模化が強く求められる気象・医療・防災を含む重要なシミュレーションの実現が困難となる。その問題解決に向けて、これらのようなシミュレーションの主要カーネル計算における、データ通信(ノード間・ノード内のデータ移動を含む)の削減・メモリ階層効率活用技術の研究を推進するのが本研究の目的である。より具体的には、対象カーネル計算として疎行列演算および格子演算を主な対象とし、研究参加者の JST-CREST プロジェクト等とも協調し、下記の研究項目について推進する。(A)疎行列計算向けメモリレイアウト・通信の最適化、(B)ステンシル計算向けアルゴリズム局所性向上。研究にあたっては、HPCI-JHPCN システムに含まれる、メモリアーキテクチャの大きく異なる複数スーパーコンピュータを用い、各アーキテクチャに適合した手法の研究を進める。

1. 共同研究に関する情報

(1) 共同研究を実施した拠点名

北海道大学
東北大学
東京大学
東京工業大学

大島聡史（東京大学）：疎行列アプリケーション最適化

岩下武史（北海道大学）：格子系アプリケーション最適化

(2) 共同研究分野

- 超大規模数値計算系応用分野
- 超大規模データ処理系応用分野
- 超大容量ネットワーク技術分野
- 超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

遠藤敏夫（東京工業大学）：全体統括
中島研吾（東京大学）：アプリケーション開発・検証
松岡聡（東京工業大学）：性能/アーキテクチャモデル
額田彰（東京工業大学）：性能/アーキテクチャモデル
長坂侑亮（東京工業大学）：疎行列アプリケーション最適化
片桐孝洋（東京大学）：性能/アーキテクチャモデル

2. 研究の目的と意義

メモリウォール問題の悪化にも関わらず、今後の大規模シミュレーションの大規模化・高性能化の進展の継続を可能とするために、主要カーネル計算における、データ通信(ノード間・ノード内のデータ移動を含む)の削減・メモリ階層効率活用技術の研究を推進するのが本研究提案の目的である。

本研究で主な対象とするメモリウォール問題や計算通信比の悪化は、ポストペタ・エクサスケールのスパコンアーキテクチャに向けて重大な問題であることが本分野の共通認識となっており、その文脈で **Communication Avoiding Algorithms** の研究に注目が集まっている。

さらには Robert Lucas らによる DOE 報告書”**Top Ten Exascale Research Challenge**” においては述べられている、主要十課題のうち、”**Memory Technology**”, ”**Data Management**”, ”**Exascale Algorithm**” の課題に、本研究が直接

対応している。特に、メモリアーキテクチャの特性に合わせたデータ構造およびアルゴリズムの研究を、近年発展が著しい疎行列データフォーマット(改良 ELL や Segmented-JDS)や、時間ブロッキング・パワーカーネルなどの、最新の研究成果をベースに行う点は重要である。さらには通信削減による消費エネルギーの削減(課題”Energy Efficiency”)およびノード間通信削減アルゴリズムを用いることによるインターコネクタアーキテクチャへのフィードバック(課題”Interconnect”)にも対処する効果が生じることが期待される。

以上のように将来のスパコンアーキテクチャにおける重大な課題に取り組む本研究の意義は極めて高いと言える。

3. 当拠点公募型共同研究として実施した意義

本研究ではメモリ・通信アーキテクチャの異なる大規模スーパーコンピュータにおける技術実証・アーキテクチャに適合した最適化を研究対象に含む。汎用マルチコアプロセッサを備える東大 FX10, GPU アクセラレータを備える東工大 Tsubame, ベクトルプロセッサを備える東北大 SX-ACE などを利用可能な JHPCN の研究として行うことにより技術をより広範囲に実証することができる。さらに研究グループとしても、東工大・東大・北大の研究者がこれまでの研究成果を基に強力に連携して研究を推進するものである。その研究分野はアプリケーションからアーキテクチャに広くまたがっており、当拠点公募型共同研究として実施する必要性が高い。

4. 前年度までに得られた研究成果の概要

(新規課題のため該当なし)

5. 今年度の研究成果の詳細

(A) 疎行列計算向けメモリレイアウト・通信の最適化

(A-1) SELL-C- σ 法に基づく行列格納手法

疎行列計算は間接参照を含むため memory-bound なプロセスである。従って疎行列演算において、演算性能と比較してメモリ転送性能の低い昨今の計算機の性能を引き出すことは困難である。係数行列の格納形式が性能に影響することは広く知られており、様々な手法が提案されている。

Compressed Row Storage (CRS) 形式は、疎行列の非零成分のみを記憶する方法である。Ellpack-Itpack (ELL) 形式は各行における非零非対角成分数を最大非零非対角成分数に固定する方法であり、実際に非零非対角成分が存在しない部分は係数=0 として計算する。CRS と比較して高いメモリアクセス効率を得られることが知られているが、計算量、必要記憶容量ともに増加する。ELL 形式を拡張し、より効率的に疎行列を記憶する手法として、Sliced ELL 形式が提案されている。Sliced ELL 形式は主として疎行列ベクトル積に使用されていたが、前進後退代入などデータ依存性を含むプロセスにも適用されている。SELL-C- σ [引用 1] は Sliced ELL を更に SIMD 向けに拡張したものである(図 4 参照)。C (Chunk Size) は SIMD 幅に相当し、非零非対角成分数の変化に応じて、 σ (Sorting Scope) を定める。図 1 の例では、長さ 2 の Chunk が 4 つで 1 コンポーネントを構成しており、SELL-2-8 と呼ばれる。非零非対角成分が変化しない場合は SELL-C-1 と呼ばれる。

本研究では Intel Xeon E5 (IvyBridge-EP) 1 ソケット (10 コア), Intel Xeon Phi (Knights Corner) 1 ノード (60 コア, 240 スレッド) を使用して検討を実施した。

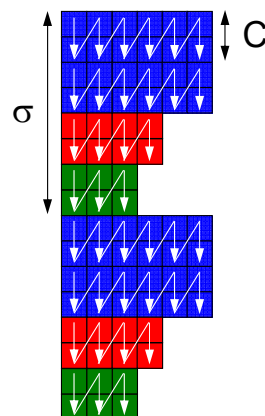


図 1 SELL-C- σ [引用 1], C: Chunk Size, σ : Sorting Scope, この図では C=2, σ =8

三次元有限要素による定常熱伝導問題から導かれる係数行列に点ヤコビ法に基づく前処理付き共役勾配法 (Preconditioned Conjugate Gradient, PCG) を適用した。図 2 は SELL-C-1 の場合の計算結果 (GFLOPS 値に基づく計算性能) である。MIC 上では C が小さいと性能が低い が 8 以上では性能はほぼ一定となり、SELL-32-1 で最高の性能が得られている。今回使用した Intel Xeon Phi 5110P (Knights Corner) は 512bit 幅の AVX512 SIMD 命令をサポートしており、64bit の倍精度実数では長さ 8 の配列に相当する。C が 8 より大きいときに最高性能を発揮するのはその関係と考えられる。最大性能は CRS の場合の 1.5 倍に相当する。一方 IvyB では C の値による影響はほとんどなく、また、CRS との差もほとんど無い。これは Intel Xeon E5-2680 v2 (Ivy-Bridge-EP) で採用されているアウト・オブ・オーダー実行の特性と考えられる [成果 3]。

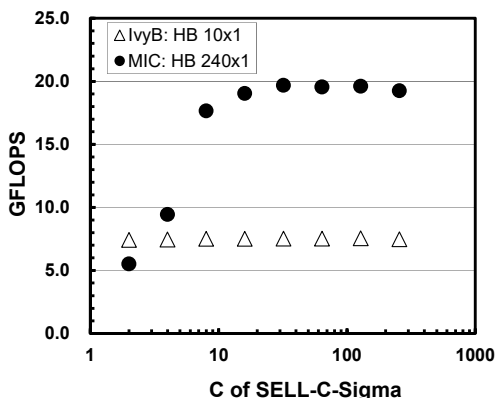


図 2 PCG 法ソルバーの性能 (総自由度数: 2,097,152=128³), SELL-C-法におけるチャンクサイズ (C) の効果

[引用 1] M. Kreutzer, G. Hager, G. Wellein, H. Fehske, and A. R. Bishop: A unified sparse matrix data format for efficient general sparse matrix-vector multiplication on modern processors with wide SIMD units. SIAM Journal on Scientific Computing 36-5 (2014) C401–C423

(A-2) 非同期型並列アルゴリズム

大規模な並列計算機を使用する場合、ノード数の増加によって通信のオーバーヘッドは増加する傾向にある。並列計算において通信は必須のプロセスであるが、通信をできる限り効率的に実施し、削減することは EFLOPS 級システムにおいて重要である。通信の削減がアルゴリズムに不安定をもたらす可能性もある。ここでは、行列解法に焦点を当てアルゴリズムに関して述べることにする。有限要素法、差分法等から得られる大規模疎行列の解法としては共役勾配法 (Conjugate Gradient, CG) に代表されるクリロフ部分空間法が広く使用されている。同手法では内積の計算が現れるが、全 MPI プロセスによる通信を必要とするため、計算量に比べて通信のオーバーヘッドの割合が大きくなる。Pipelined CG 法 [引用 2] では、CG 法の結果が変わらないように計算順序を変更し、MPI-3 でサポートされている非同期集団通信 (MPI_Iallreduce) によって、集団通信のオーバーヘッドを隠蔽することに成功している。

図 3 は 1,474,560 自由度 (491,520 節点=96 × 80 × 64) の三次元静弾性問題 (有限要素法) から導かれる疎行列を並列 ICCG 法で解いた場合の、最大 64 ノード (1,280 コア, 1,280 MPI プロセス) までの加速率の結果である。1 ノードあたり 2 ソケット、各ソケット 10 コアの Intel Xeon E5-2680 v2 (Ivy-Bridge-EP) によるクラスタを使用している。適用したアルゴリズムは:

- Alg.1: オリジナルの ICCG 法
- Alg.2: Chronopoulos/Gear アルゴリズムによる ICCG 法 [引用 2]
- Alg.3: Pipelined CG 法 [引用 2]
- Alg.4: Gropp アルゴリズムによる CG 法 [引用 2]

である。Alg.1 の 1 ノード (20 コア) における ICCG 法の計算時間から得られる性能を 20 とした場合の各アルゴリズムによる速度増加率が示されている。内積部分には非同期集団通信 (MPI_Iallreduce) を適用している。Alg.2 は CG 法中の内積を一箇所に集めることによって、集団通信の回数を減らしているだけであるが、64 ノード (1,280 コア) ではオリジナル手法との明瞭な差が認められる。Alg.3, Alg.4 でも同様であり、Alg.4 が最も高い性能を示していることがわかる。

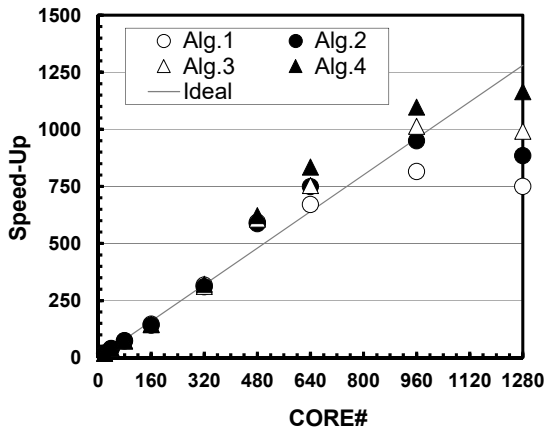


図 3 三次元静弾性問題(有限要素法)における並列 ICCG 法の Ivy-Bridge クラスタ上での計算性能, 1,474,560 自由度 (491,520 節点=96 ×80×64), 最大 64 ノード(1,280 コア, 1,280MPI プロセス)までの加速率

[引用 2] P. Ghysels and W. Vanroose, Hiding global synchronization latency in the preconditioned Conjugate Gradient algorithm, Parallel Computing 40-7, 224-238, 2014

(A-3) 悪条件問題向け並列前処理手法

不均質な物性を有する悪条件問題に領域分割に基づく並列前処理付き反復法を適用した場合の性能と安定性について評価を実施した。前処理手法は BILUT(p,d,t)法に基づく手法であり、領域分割手法としては LBJ(Localized Block Jacobi)及び HID (Hierarchical Interface Decomposition)について検討した。提案手法は Hetero3D(並列有限要素法による三次元固体力学問題を対象としたベンチマーク)に実装され, Fujitsu PRIMEHPC FX10 (Oakleaf-FX) (東大情報基盤センター)の 240 ノード, 3,840 コアまで良好な安定性とスケーラビリティが得られた。一般的に LBJ よりも HID の方が幅広いパラメータ値に対してやや安定である(図 4)。

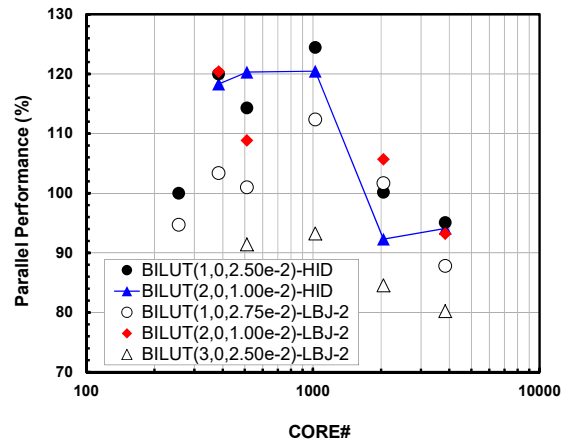
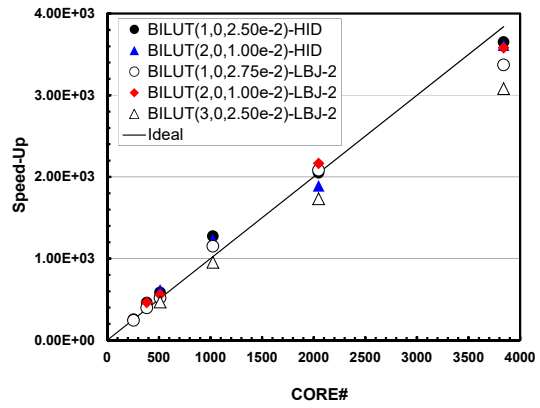


図 4 並列 BILUT(p,d,t)-LBJ/HID-GPBi-CG 法の性能 (Strong Scalability), 前処理行列 set-up + 線形ソルバー計算時間, Oakleaf-FX 上における各手法の t の最適値を適用, BILUT(1,0,2.5 × 10⁻²)-HID256 コアにおける計算時間によって無次元化, 400 × 320 × 240 節点 (=3.072 × 10⁷ 節点, 30,420,159 要素, 9.216 × 10⁷ DOF) (上)速度増加率, (下)並列性能

(A-4) AMB 疎行列フォーマット

疎行列データの格納には通常メモリ使用量と演算量の削減を目的として非ゼロ要素のみを保持する疎行列フォーマットを使用する。代表的なものとして COO や CSR フォーマットがあるが、これらを含む現実的な疎行列フォーマットでは疎行列ベクトル積計算時にベクトル要素への間接参照が必要となり、引き起こされるランダムアクセスによってキャッシュメモリでのヒット率が低くなる。より小さなキャッシュ容量しか持たない GPU ではより深刻な問題となっている。また、疎行列フ

フォーマットでは各非ゼロ要素に関して値だけでなく行や列のインデックスを保持する必要がある。その結果、行列ベクトル積計算そのものが持つメモリバウンドであるという性質がより悪化するため、性能がメモリバンド幅に律速し低下し、GPU などの持つ高い演算性能を十分に活用出来ていないという問題がある。

疎行列ベクトル積計算の高速化を目的として、新たな疎行列フォーマットである Adaptive Multi-level Blocking (AMB) フォーマットの構築を行った。対象とする疎行列データに対して、多段階の分割や圧縮処理を行うことによって、ベクトル要素へのアクセスの局所性向上と非ゼロ要素のインデックス要素の圧縮によるメモリバウンド性の緩和を実現した。高速な疎行列ベクトル積を実現する既存の疎行列フォーマットである SELL-C- σ と疎行列ライブラリである cuSPARSE と yaSpMV との性能評価を行った結果を図 5 に記す。

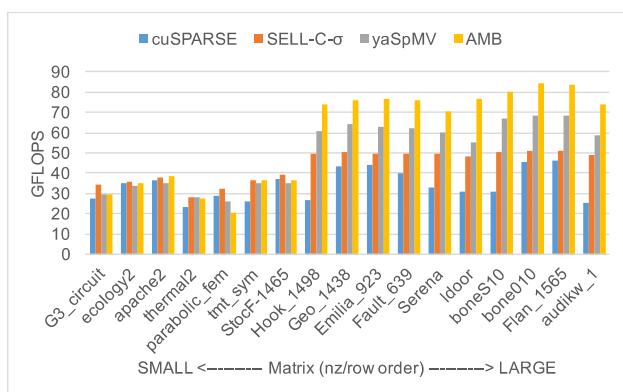


図 5. 疎行列フォーマット毎の疎行列ベクトル積計算性能

多種多様な疎行列データに対して評価を行った結果、AMB フォーマットを用いることによって行あたりの平均非ゼロ要素数が大きい行列データに対して、高い疎行列ベクトル積計算性能を達成することが可能であることを示した。また、同データに対して、CSR と比較した各手法でのメモリアクセス量の比率を図 6 に示す。

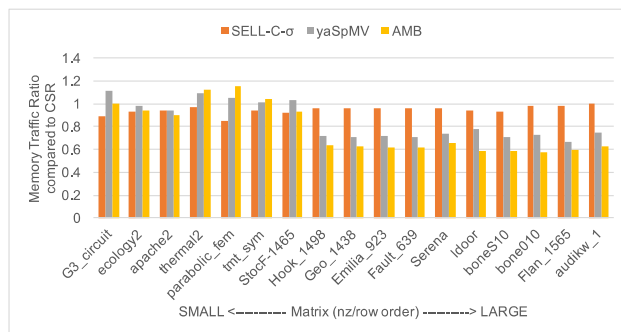


図 6. 疎行列ベクトル積計算時のメモリアクセス量の比率

AMB フォーマットを用いることによって、既存の疎行列フォーマットを用いた場合から大幅にメモリアクセス量を削減することが可能であることを示している。また、メモリアクセス量を適切に削減出来ている疎行列データにおいて疎行列ベクトル積計算性能が向上していることから、疎行列ベクトル積計算における通信削減の重要性を示していると言える。

GPU で高い効果を発揮した AMB フォーマットであるが、メモリアクセス量削減という観点からは他のプロセッサアーキテクチャにも有効であると考えられる。そこで GPU と同様の高いメモリバンド幅を備える SX-ACE において評価を行ったが顕著な性能向上は得られなかった。このような結果になった原因は幾つか考えられるが、基本的にはプロセッサアーキテクチャの差によるものである。まず AMB フォーマットで用いる 16bit 整数の使用によるメモリアクセス量の削減であるが、SX-ACE では整数の場合 32bit または 64bit しかベクトル命令が用意されていない。また SX-ACE ではベクトル長が 256 のベクトルパイプラインであり、基本的には 256 要素の全てが揃わないと後続の処理へ進まないため一部の要素が ADB のキャッシュミスを生じた場合のペナルティが大きい。また GPU のようにたくさんのスレッドが同時に実行することでレイテンシを軽減するような効果もない。

(B) ステンシル計算向けアルゴリズム局所性向上
(B-1) 実アプリケーションの局所性向上・大規模化

局所性向上・通信削減によりメモリ階層を効率利

用するアルゴリズムの研究として、ステンシル演算の時間ブロッキングに関する研究を取組んでいる。GPU のもつデバイスメモリ階層を考慮するために、東工大 TSUBAME スパコンを主プラットフォームとして研究に取り組んだ。時間ブロッキングは、ステンシル対象領域のうち一部領域(上位メモリに収まる範囲)の計算にとりかかったら、その一部領域について複数時間ステップの計算を一気に行う手法である。これは局所性向上手法としてかつてより知られた方法であるが、ここではキャッシュ効率向上よりも異なるメモリ階層間のデータ移動を削減するために用いている。

一般的に時間ブロッキングをステンシル計算に基づいた既存のアプリに組み込むには、ループ構造の大幅な改変のために、高いプログラミングコストを必要とする。これを(完全な解消ではないが)大きく軽減するために、代表者らが提案・実装しているメモリ階層活用ランタイムライブラリである HHRT (Hybrid Hierarchical RunTime)を用いている。時間ブロッキングを比較的低いプログラミングコストでアプリコードに組み込み、HHRT 上で動作させるというアプローチを取る。

この手法を、MPI および CUDA で実装された既存の大規模シミュレーションコードである、都市気流シミュレーションと樹状結晶シミュレーション(いずれも東工大青木尊之らによる)に適用した。コードの一部改変により、時間ブロッキングおよび関連の最適化をほどこした。結果として、これまでデバイスメモリ容量にステンシル配列サイズが制限されていたアプリケーションを、その制限を超えて実行することに成功した。このときの性能は、図 7 および図 8 に示すように、アプリの用いる格子配列が GPU デバイスメモリ容量を超えても動作可能であることを実証した。性能へのオーバーヘッドはデバイスメモリ容量の 4 倍時に約 20%~30% である。

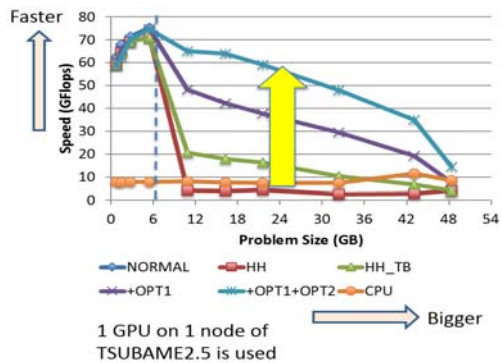


図 7 都市気流シミュレーションに時間ブロッキングを組み込み HHRT 上で実行した場合の性能

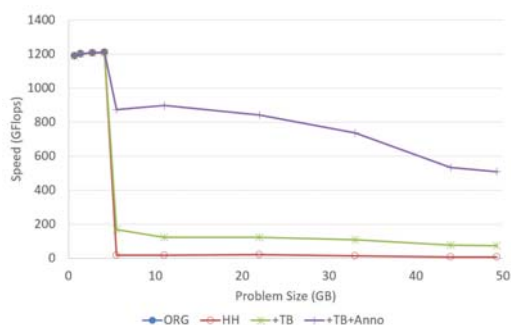


図 8: 樹状結晶シミュレーションに時間ブロッキングを組み込み HHRT 上で実行した場合の性能

(B-2) Flash メモリ階層対応とアーキ検討

上記の成果を拡張し、適切なメモリ階層設計を持つスパコンアーキテクチャ上において、ステンシル計算において実現可能な問題規模をさらに拡大する試みを行った。このためには、ホストメモリ(DDR 系 DRAM)よりもさらに大容量としやすい Flash SSD を第三のメモリ階層ととらえる。このようなデバイスは、HPCI-JHPCN システムの中では TSUBAME にいち早く導入されていた。その対応のための変更を HHRT ライブラリにほどこした。より具体的には、上位メモリ階層のユーザデータのスワップ先として、ホストメモリだけでなく Flash SSD を対象に含めた。

このような研究開発を TSUBAME 上で行っていたが、予備実験において、TSUBAME のもつ Flash SSD (2010 年に導入されたもの)のアクセス性能が、この目的には不足することが分かった。

そのため、HPCI-JHPCN システム外ではあるが、

最新の m.2 SSD を搭載する PC サーバを用意し、その上で評価を行った。TSUBAME2 の SSD の性能が Read 300MB/s 程度、Write 200MB/s であったのに対し、この m.2 SSD は Read 2.4GB/s, Write 1.5GB/s と、10 倍近い高速アクセスを可能とする。このデバイス利用と、時間ブロッキングパラメータの詳細な解析により、図 9 に示すような速度を達成した。このとき、デバイスメモリ 12GiB およびホストメモリ 64GiB を超え、最大 192GiB のケースで実行成功している。96GiB の際の性能低下は 45%程度であり、これは無視できないオーバーヘッドではあるが、計算資源が大きく制限されるときでも、大規模計算を現実的な速度で可能であることを示した。

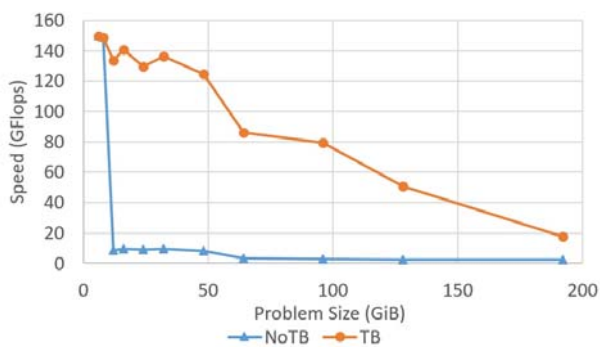


図 9: Flash 対応 HHRT 上において、7 点ステンシル+時間ブロッキングを実行した場合の性能

6. 今年度の進捗状況と今後の展望

(A) 疎行列計算向けメモリレイアウト・通信の最適化

(A-1) SELL-C- σ 法に基づく行列格納手法

SELL-C- σ 法はこれまで専ら疎行列ベクトル計算に適用されており、本研究においても同様である。ICCG 法等の前処理付き反復法のデータ依存性を含むプロセス(不完全分解, 前進後退代入)への適用は Gauss-Seidel 反復法を除いてはまだ実施されておらず、早急に取り組む必要がある。

(A-4) AMB 疎行列フォーマット

AMB フォーマットではメモリアクセス量の予測とそれに基づく実行時間モデル化が非常に高い精度で実現されている。図 3 にフォーマット変換後の予測メモリアクセス量と実行時間の対応を示す。

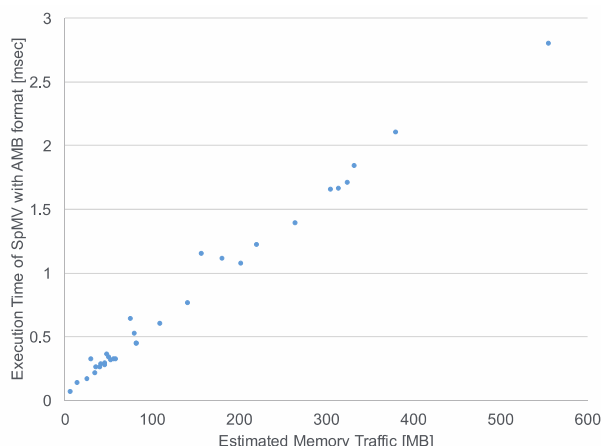


図 9: フォーマット変換後の予測メモリアクセス量と実行時間の対応

多くの疎行列データに対して、メモリアクセスの予測量に比例して疎行列ベクトル積計算の実行時間が推移している。AMB フォーマットの最適化の一つである行列の列分割により入力ベクトルへのランダムアクセス時のキャッシュミス率を抑えた結果、不確定要素が除去されモデルの精度が高くなっている。モデルによって実行時間を推定可能というこの事実はアプリケーションをマルチGPU, マルチノードに拡張した時に重要性を増す。並列実行時には一番実行時間が長いGPUが律速となり全体の性能を制限するため、GPU間のロードバランスが最も重要である。AMB フォーマットを使用するのであれば、通信時間やBLASルーチン等のモデル化が容易な処理に加えて疎行列ベクトル積計算部分も含めたソルバーなど全体のモデル化によりロードバランスの最適化が可能となる。

(B) ステンシル計算向けアルゴリズム局所性向上 (B-1) 実アプリケーションの局所性向上・大規模化

ステンシル計算の問題規模は、通常では高速かつ小容量のメモリ階層によって限定される。これに対して、アルゴリズム的な局所性向上(店舗らるブロッキング)とメモリ階層の隠蔽を行うミドル

ウェア (HHRT ライブラリ) の組み合わせにより、比較的小さいオーバーヘッドでその限界を超えることができることを、実際のシミュレーションコードを用いた TSUBAME スパコン上の性能評価を通して示した。

(B-2) Flash メモリ階層対応とアーキ検討

デバイスメモリ・ホストメモリに加えたメモリ階層として Flash SSD を用いることによるさらなる問題規模拡大について取り組んだ。この目的のためには、TSUBAME スパコンの SSD よりも最新の GB/s 級のアクセス速度を持つデバイスの搭載が必要であることを示した。

今後は、(B-1) と (B-2) の成果の統合により、Flash SSD の規模における実シミュレーションコードの評価が急がれる。さらには将来のスパコンに向けて、今回アーキテクチャのメモリ階層の詳細評価を通して判明した、高速 Flash デバイスの次世代スパコンへの導入について、種々のトレードオフを考慮しつつ検討する。

7. 研究成果リスト

(1) 学術論文

(2) 国際会議プロシーディングス

Toshio Endo, Yuki Takasaki, Satoshi Matsuoka. Realizing Extremely Large-Scale Stencil Applications on GPU Supercomputers . In Proceedings of The 21st IEEE International Conference on Parallel and Distributed Systems (ICPADS 2015), Melbourne, December, 2015.

Nakajima, K., Parallel Iterative Solvers for Ill-conditioned Problems with Heterogeneous Material Properties, Proceedings of IHPACES 2016 (Sixth International Workshop on Advances in High-Performance Computational Earth Sciences: Applications and Frameworks)

in conjunction with ICCS 2016, San Diego, CA, USA, 2016 (in press)

Yusuke Nagasaka, Akira Nukada and Satoshi Matsuoka. “Adaptive Multi-level Blocking Optimization for Sparse Matrix Vector Multiplication on GPU”, In Proc. of the 16th International Conference on Computational Science (ICCS 2016), Procedia Computer Science series, Elsevier, June 2016, to appear.

(3) 国際会議発表

Toshio Endo, Satoshi Matsuoka. Realizing Extremely Large-Scale Stencil Applications on GPU Supercomputers with a Memory Hierarchy Management Runtime Library. Workshop on Programming Abstractions for Data Locality (PADAL 2015), Berkeley, June 25, 2015.

Yusuke Nagasaka, Akira Nukada, Satoshi Matsuoka. “Multi-Level Blocking Optimization for Fast Sparse Matrix Vector Multiplication on GPUs”, In Proc. of 2015 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, Austin, TX., Technical Poster, ACM Press, Nov. 2015.

Nakajima, K., Parallel Iterative Solvers based on Domain Decomposition with Robust Preconditioners for Ill-conditioned Problems, IPSJ Proceedings of the 2nd Annual Meeting on Advanced Computing System and Infrastructure (ACSI2016), 2016

Yusuke Nagasaka, Akira Nukada, Satoshi Matsuoka. “Fast Sparse Matrix Vector Multiplication with Highly-Compressed Sparse

Format”, GPU Technology Conference 2016,
poster, April 2016.

(4) 国内会議発表

高寄 祐樹, 遠藤 敏夫, 松岡 聡. GPU クラスタ
における大規模都市気流シミュレーションの最適
化と性能モデル. 情報処理学会ハイパフォーマンス
コンピューティングと計算科学シンポジウム
(HPCS2015), 東京, 2015 年 5 月 20 日.

長坂 侑亮, 額田 彰, 松岡 聡. 「多段階ブロッキング
によるメモリアクセス量削減を図った GPU 向け
疎行列ベクトル積計算手法の性能評価」, NVIDIA
GTC Japan 2015, ポスター, 2015 年 9 月.

長坂 侑亮, 額田 彰, 松岡 聡. 「疎行列ベクトル
積計算を対象とした GPU 向けメモリアクセス削
減手法」, 情報処理学会研究報告, Vol.
2015-HPC-151, No. 8, 2015 年 9 月.

中島 研吾, 片桐 孝洋, 大島 聡史, 埜 敏博,
ppOpen-APPL/FVM を使用した並列有限要素法
アプリケーションの開発, 情報処理学会研究報告
(HPC-151-24), 2015

遠藤 敏夫. 大規模・高性能演算のための多階層
メモリの活用 . 情報処理学会研究報告,
2015-HPC-153 No.14, 7pages, 2016 年 3 月 2 日.

(5) その他 (特許, プレス発表, 著書等)