Vol.2022-ARC-248 No.28
Vol.2022-SLDM-198 No.28
Vol.2022-EMB-59 No.28
2022/3/10

疎行列演算高速化のためのメモリアーキテクチャ探索

萩原 汐[1,a)]　児玉 宏喜[1]　吉川 隆英[1]　幸 朋矢[2]　遠藤 敏夫[2]

概要：近年のハイパフォーマンスコンピューティングでは、演算速度よりデータ転送速度がボトルネックとなってきている．ボトルネックを解消するには，ハードウェアによる手法とソフトウェアによる手法を適切に組み合わせる必要があるが，どのように組み合わせれば効果的に高速化できるかは明らかではない．特に，疎データを扱うワークロードではメモリアクセスにランダム性があることから解析的に実効性能を求めることが困難であり，高速化の効果を定量的に見積もることは難しい．東京工業大学が開発しているPMNet (Performance predictor of Memory Network) では，実際にワークロードを実行した際のトレース結果を使って，任意のメモリアーキテクチャにおける実行時間を推定する．PMNet を使えば，メモリアクセスを考慮したメモリアーキテクチャの性能を高速に見積もることができる．本報告の目的は，疎行列演算を扱うワークロードの実効性能を向上させるための設計指針を立てることである．まず，疎行列ベクトル積 (SpMV; Sparse Matrix-Vector multiplication) を使った PMNet の精度検証を行い，目的達成に十分な精度をもつことを示す．更に，IRDS ロードマップのデバイスパラメタを使って，PMNet がターゲットとしている 2028 年に実現可能なプロセッサにおける SpMV の性能を推測すると共に，高速化のための設計指針を検討する．

キーワード：フォン・ノイマン・ボトルネック, 疎行列ベクトル積, SpMV, 性能予測

Memory architecture exploration for sparse matrix-vector multiplication

Shiho Hagiwara[1,a)]　Hiroyoshi Kodama[1]　Takahide Yoshikawa[1]　Tomoya Yuki[2]　Tosio Endo[2]

Abstract: In recent high-performance computing, data transfer speed has become the bottleneck rather than computation speed. The solution to this problem can be hardware or software. However, it is unclear how to combine them to achieve effective speedup. In particular, for sparse workloads, the randomness of memory access makes it difficult to estimate the computational performance analytically. PMNet (Performance prediction of Memory Network), developed by Tokyo Institute of Technology, estimates the performance of any memory architecture using the memory trace results of actual workload execution. It can quickly evaluate the performance of a memory architecture while considering memory accesses. This report aims to provide design guidelines for improving the performance of sparse matrix operations. First, we conduct accuracy verification using sparse matrix-vector multiplication (SpMV) workloads and show that PMNet has enough accuracy for our aims. Next, we estimate SpMV performance on 2028's processors using PMNet and the device parameters of the IRDS roadmap and discuss design guidelines to accelerate it.

Keywords: von Neumann bottleneck, Sparse matrix-vector multiplication, SpMV, performance estimation

1. はじめに

近年のハイパフォーマンスコンピューティングでは，プロセッサへのデータ転送が間に合わず，演算速度が遅く

[1] 富士通株式会社
Fujitsu ltd.
[2] 東京工業大学
Tokyo Institute of Technology
[a)] hagiwara.shiho@jp.fujitsu.com

Vol.2022-ARC-248 No.28
Vol.2022-SLDM-198 No.28
Vol.2022-EMB-59 No.28
2022/3/10

なってしまう，フォン・ノイマン・ボトルネックが課題となっている．この問題はメモリアクセス性能が問題になりやすい疎行列演算において顕著に現れる．例えばスーパーコンピュータの世界最高性能を競う TOP500 では密行列を扱う HPL において，第 1 位の「富岳」は 82.3 %, 第 2 位の Summit で 74.0 % の実行効率を達成できるのに対し，疎行列を扱う HPCG においては第 1 位の「富岳」でも 3.0 %, 第 2 位の Summit では 1.5 %に留まってしまう [1,2].

この課題に対応する方策として，まずソフトウェア側として，アルゴリズムやコードのチューニングによってワーキングセットのサイズを調整し，メモリを効率良く使用する手法や [3]，プリフェッチによって必要になりそうなデータを予めキャッシュに読み込んでおく手法 [4] などがある．更にハードウェア側として，HBM や 3D SRAM などを活用したラストレベルキャッシュの大容量化，ローカルメモリの利用，GDDR のようなメモリ帯域の増強などの手法がある．

これらの手法はそれぞれに長所と短所があるため，それぞれの効果を見積もりながら，適切に組み合わせる必要がある．しかしながら，キャッシュの振る舞いは複雑であるため，机上計算だけで各手法の効果を見積もることは非常に難しい．そこで，キャッシュ動作を意識したメモリ階層シミュレーションを実施し，ソフトウェアの最適化による適切な階層へのメモリ配置，並びに，アプリケーションの性質を踏まえたメモリ階層の最適設計が必要になる [5,6].

東京工業大学が開発した PMNet (Performance predictor of Memory Network) [7] は，内部にキャッシュのシミュレータを備えた性能推定ツールである．実アプリケーション実行時に採取できるメモリアクセストレースを用いて，様々なメモリ構成 (階層・容量・バンド幅) に対してアプリケーション実行時間を推定する．PMNet はプロセッサ内部のサイクル精度のシミュレーションは行わず，データの移動を中心としてキャッシュの振る舞いとデータ処理時間だけに注目をして性能推定を行う．精度は犠牲になるが，高速に性能を推定できる．

本稿では，この PMNet の精度ならびに実用性の評価を行う．具体的には様々な疎行列データを使った疎行列ベクトル積 SpMV (Sparse Matrix-Vector multiplication) について評価することで，PMNet がメモリアーキテクチャ設計やソフトウェアチューニングに有効であるかの検証を行う．更に，検証結果を踏まえ疎行列演算高速化のためのチューニングへの PMNet の活用方法を提案する．

本稿の構成は以下の通りである．まず 2 章では，現行アーキテクチャを想定し，疎行列処理で頻出する SpMV の性能を机上計算し，実測値と比較することで机上計算による性能推定が困難であることを示す．3 章では，同様の性能推定を PMnet で行い，その精度を検証する．4 章では，PMNet がターゲットとしている 2028 年に実現可能なメモリ構成パラメタ (容量・バンド幅) での性能推定を PMNet で行い，その結果を踏まえてハードウェア構成を変更する場合には容量とバンド幅のどちらを優先して強化すべきか，ソフトウェアを最適化するにはどの階層のキャッシュにどのようなデータを配置すれば良いのかの指針を示す．

表 1 CPU 演算性能

倍精度浮動小数点演算性能	18.4 GFLOPS
時間当たり命令数 (IPS)	18.4 GIPS

表 2 キャッシュメモリおよびメインメモリ諸元

	capacity	read bandwidth	write bandwidth
L1	32 KiB	294.4 GB/s	147.2 GB/s
L2	1 MiB	147.2 GB/s	
L3	1.375 MiB	147.2 GB/s	
Memory	96 GiB	{15.8, 31.6, 47.4} GB/s	

L1/L2/L3 はコア当たり．Memory は CPU 当たり．

2. SpMV 実効性能の机上計算の精度

本章では，現行アーキテクチャを想定し，SpMV 実行時の時間当たり浮動小数点演算回数 (FLOPS; FLoating point number Operations Per Second) を机上計算により推定し，その値を実測と比較する．なお，FLOPS 値は CPU のハードウェア仕様値として用いられる理論値 (演算器数 × ビット幅 × 周波数で与えられる数値) と，実際にワークロード実行して計測して得られる値の 2 種類があるが，本報告では前者を演算性能，後者を実効性能と表記する．つまり，本章で机上計算するのは実効性能である．

実測は Intel®Xeon®Gold 5218 (16 コア) 1 ソケットで行ったため[*1]，机上計算の条件も同 CPU に合わせる．使用したパラメタ値は表 1, 2 の通りである．時間当たり実行命令数 (IPS; Instructions Per Second) の正確な値を知ることは困難であるが，プロセッサ・アーキテクチャ等を参考に決めた．

なお，L1, L2 キャッシュはコア毎に独立してもつが，L3 キャッシュは共有キャッシュであるため (図 4)，容量は 1.375 MiB×16 = 22 MiB になる．また，メモリのバンド幅は STREAM benchmark [8] で測定した値を使用した．メモリバンド幅は 3 つ記載しているが，本章で使用する値は 47.4 GB/s である．

2.1 SpMV 性能の机上計算

SpMV では疎行列 A とベクトル $\mathbf{x}$ の積を計算する．

$$\mathbf{y} = A \cdot \mathbf{x} \tag{1}$$

疎行列 A を CSR (Compressed Row Storage) 形式で格納した場合のコード例を以下に示す．

```
for (i=0; i<nrow; i++){
```

[*1] ハイパースレッドを無効化し 16 スレッドで実行

Vol.2022-ARC-248 No.28
Vol.2022-SLDM-198 No.28
Vol.2022-EMB-59 No.28
2022/3/10

```
        for (j=rpt[i]; j<rpt[i+1]; j++)
            y[i] += val[j] * x[ind[j]];
}
```

ここで，y および x は浮動小数点型の配列でそれぞれベクトル $\mathbf{y}, \mathbf{x}$ に対応する．疎行列 A は整数型配列 ind[j] と浮動小数点型配列 val[j] に格納されており，それぞれ列番号と数値に対応する．行列データ 12B，ベクトルデータ 8B，合計 20B のデータを使って，2 回の浮動小数点演算が行われることがわかる．

コア当たりの浮動小数点演算性能が $FLOP_{\mathrm{core}}$ [回/cycle] だとすると，CPU で演算し続けるためには

$$\frac{12+8}{2/FLOP_{\mathrm{core}}} = 10FLOP_{\mathrm{core}}\,[\mathrm{B/cycle}] \tag{2}$$

のデータ供給が必要になる．

Intel®Xeon®Gold 5218 プロセッサの場合，AVX-512 FMA ユニットを 1 つ搭載しているため $FLOP_{\mathrm{core}} = 8$ であることから [9]，80 B/cycle のデータ供給が必要である．しかし各キャッシュメモリのデータのバンド幅は コア当たり L1 128 B/cycle, L2 64 B/cycle, L3 64 B/cycle である [10]．データが L1 キャッシュにあればデータ供給は演算に間に合うが，そうでなければ処理速度はデータ転送速度に律速される．

L1 キャッシュに全データが格納されることはほとんどないと考えられるので，処理速度は L2 以降のバンド幅に律速されることになる．ただし，データの一部が L1 キャッシュに格納されていれば，L2 以降から転送すべきデータ量は小さくなりデータ供給は間に合う．これが実現する条件は，L1 キャッシュミス率 $p_{\mathrm{v},1}$ について，

$$\frac{(12+8p_{\mathrm{v},1})\times FLOP_{\mathrm{core}}}{2} \le 64\,[\mathrm{B/cycle}]. \tag{3}$$

よって $p_{\mathrm{v},1} \le 0.5$ を満たせば，L2 キャッシュおよび L3 キャッシュのデータ供給能力は不足しない．

L3 キャッシュに全データが格納されている場合は，$p_{\mathrm{v},1} \le 0.5$ を満たすか否かでボトルネックが変わる．

(i) $p_{\mathrm{v},1} \le 0.5$ の場合　データ転送速度に律速されることなく，コアの演算性能で SpMV を実行できる (本稿ではこの状態を「演算ボトルネック」と呼ぶ)．

(ii) $p_{\mathrm{v},1} > 0.5$ の場合　L2 および L3 がボトルネックとなる．このときの実効性能は，

$$\frac{2}{(12p_{\mathrm{m},1}+8p_{\mathrm{v},1})/BW_{\mathrm{L2}}} \tag{4}$$

と計算できる．条件から $0.5 < p_{\mathrm{v},1} \le 1$，また行列データは再利用がないため $p_{\mathrm{m},1} = 1$，であることから，実効性能は $BW_{\mathrm{L2}}/10 \sim BW_{\mathrm{L2}}/8$ となる．

L3 キャッシュに格納されていないデータがある場合，メインメモリから取得する必要がある．メインメモリのバンド幅は L2, L3 と比較すると桁違いに小さいため，このときの実効性能はメモリからのデータ供給速度に律速される．行列データの L3 キャッシュミス率を $p_{\mathrm{m},3}$，ベクトルデータの L3 キャッシュミス率を $p_{\mathrm{v},3}$ として，このときの実効性能を求めると，2 回の浮動小数点演算にかかる時間がメインメモリからのデータ転送時間に相当するので，

$$\frac{2}{(12p_{\mathrm{m},3}+8p_{\mathrm{v},3})/BW_{\mathrm{mem}}} \tag{5}$$

となる．

SpMV の実効性能を机上計算により導出するには，キャッシュミス率を知る必要があることから，困難であることがわかる．しかしながら，キャッシュメモリよりもメインメモリの方がバンド幅が小さいことを考えると，実効性能は最低で $BW_{\mathrm{mem}}/10$ FLOPS 程度になることは予想できる．

2.2　机上計算の精度検証

ここでは，HPCG でも使われている有限要素法から得られる疎行列 (以下 HPCG の疎行列) および SuiteSparce Matrix Collection [11] から取得した 12 の疎行列を使用して，2.1 章で行った机上計算の精度を実測との比較により検証する．使用した疎行列は [12] で使用された 14 の行列から，密行列と正方行列でない行列を除いたものである．一覧を表 3 に示す．

実効性能の実測値は　2 章に示したコード部分の実行時間 (10 回の平均値) から時間当たり浮動小数点演算数 (FLOPS) を計算することで求めた．このとき，浮動小数点演算数は疎行列の非ゼロ数 ×2 とした．

2.2.1　HPCG の疎行列を使った精度評価

HPCG の疎行列の非ゼロ位置は図 1 のように規則的になっている．ほとんどの行について，非ゼロ位置は前行の 1 要素ずれになっていること，非ゼロは 3 要素ずつ連続していることから，ベクトルデータの内約 3 分の 2 は L1 に存在すると考えられる．よって，$p_{\mathrm{v},1} \simeq 1/3$ となり，前章 (i) の場合に相当する．

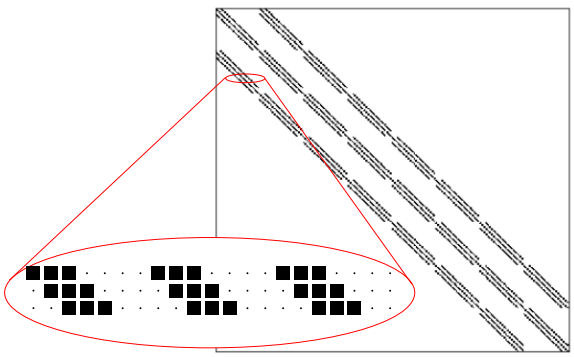

図 1　HPCG の疎行列の非ゼロ位置

データサイズが L3 キャッシュ容量を下回る場合は演算

ボトルネックになるので，実効性能は，コア当たり演算性能 × コア数で

$$18.4\ [\mathrm{GFLOPS/core}] \times 16\ [\mathrm{cores}] = 294.4\ [\mathrm{GFLOPS}] \quad (6)$$

と計算できる．

データサイズが L3 キャッシュ容量を上回ると，実効性能は式 (5) で表される．簡単のため，ベクトルデータのみが L3 キャッシュに格納されているとすると，$p_{\mathrm{v},3} = 0$, $p_{\mathrm{m},3} = 1$ より，

$$\frac{2}{12/47.4} = 7.9\ [\mathrm{GFLOPS}] \quad (7)$$

と計算できる．

図 2 に HPCG の疎行列を入力行列とした場合の実測評価結果と机上計算結果の比較を示す．格子数を増減することで様々なサイズの行列を得られるので，$8^3 \times 8^3, 16^3 \times 16^3, ..., 160^3 \times 160^3$ の行列サイズで推定および実測を行い，行数変化による実効性能の変化を表わすグラフとした．行数が $42^3 \simeq 7.4 \times 10^4$ のとき，データサイズが L3 キャッシュ容量を超え，机上計算値は 294.4 GFLOPS (式 (6)) から 7.9 GFLOPS (式 (7)) に変わる．図 2 では机上計算値の最大値が縦軸の最大値を超えるため，一部表示されていない．実測のピーク性能は 30.2 GFLOPS であり，机上計算値を大きく下回る結果となった．

机上計算では，メインの演算処理 (2.1 章のコードの最内ループにある積和演算) しか考慮していないが，実際には，他スレッドと同期をとるための命令等の様々な命令を実行する必要がある．また，キャッシュミスによるストール，利用資源の競合等により実効 IPS が下がることでも，実行時間は長くなる．このような理由により，理論性能と実測性能に大きな差が出る．

また，データサイズが L3 キャッシュ容量を超えた場合，式 (7) では一律に $p_{\mathrm{m},3} = 1$ となることを仮定したが，実際には，行数 5×10^5 以下のサイズでは L3 キャッシュに行列データも格納されているため $p_{\mathrm{m},3} < 1$ となる．このため，実測では式 (7) を上回る性能になったと考えられる．

2.2.2 SuiteSparce Matrix の疎行列を使った評価

一般的な疎行列を対象とする場合は，キャッシュミス率を推定することが困難であるため，本章では範囲の形で推定する．

データサイズが L3 キャッシュ容量より小さい場合，L1 キャッシュミス率が $p_{\mathrm{v},1} > 0.5$ であれば式 (4) より 14.7-18.4 GFLOPS，$p_{\mathrm{v},1} \leq 0.5$ であれば演算ボトルネックとなり 294.4 GFLOPS と計算できる．データサイズが L3 キャッシュ容量より大きい場合は，キャッシュミス率を $0 \leq p_{\mathrm{v},3} \leq 1, p_{\mathrm{m},3} = 1$ とすると，式 (5) より 4.74-7.90 GFLOPS と計算できる．

図 3 に各疎行列のデータサイズと SpMV 性能の実測結果と机上計算値を示す．点線は L3 キャッシュサイズ 22MB を

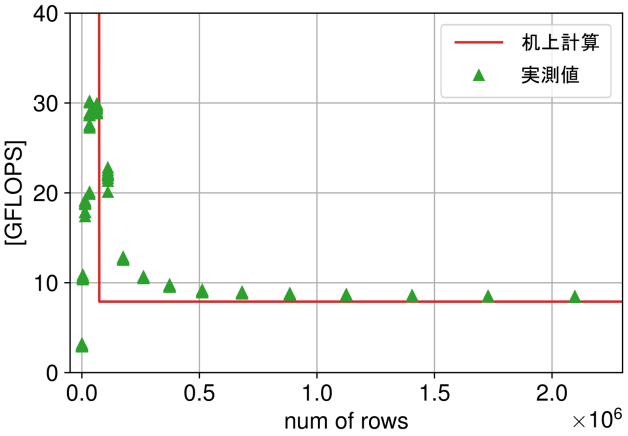

図 2 HPCG の疎行列を使った場合の机上計算の精度評価

表 3 評価に使用した疎行列一覧

name	kind	ncol	nnz
scircuit	Circuit Simulation	170998	3.28e-5
mac_econ _fwd500	Economic Problem	206500	2.99e-5
conf5_4	Theoretical/Quantum Chemistry Problem	49152	7.93e-4
mc2depi	2D/3D Problem	525825	7.60e-6
rma10	Computational Fluid Dynamics Problem	46835	1.08e-3
cop20k_A	2D/3D Problem	121192	1.79e-4
webbase-1M	重み付き有向グラフ	1000005	3.11e-6
cant	2D/3D Problem	62451	1.03e-3
pdb1HYS	重み付き無向グラフ	36417	3.28e-3
consph	2D/3D Problem	83334	8.65e-4
shipsec1	Structural Problem	140874	3.94e-4
pwtk	Structural Problem	217918	2.45e-4

示しており，この線より左側 (scircuit と mac_econ_fwd500) は L2/L3 キャッシュもしくは演算が，その他はメモリがボトルネックと考えられる．前述の計算結果と比較すると，ほぼ全て (cop20k_A 以外) の疎行列で，実測値は机上計算の範囲から外れるという結果になった．

外れた原因としては，まず，scircuit と mac_econ_fwd500 については演算ボトルネックだったために，前章と同じ原因で大きな差が出た可能性がある．また，その他の疎行列 (メモリボトルネックである疎行列) については，机上計算範囲を上回ったものはキャッシュミス率 $p_{\mathrm{m},3} = 1$ の推定が悲観的過ぎたこと，下回ったものはキャッシュミスによるストールの影響等によるプロセッサの実効 IPS の低下等が原因として考えられる．

以上から，一般的な疎行列について SpMV の性能を精度良く推定することは困難であることがわかる．

3. PMNet を用いた性能予測の精度

前章で示した通り，机上計算ではキャッシュミス率を特定することが困難であるために，精度良い性能予測が難し

Vol.2022-ARC-248 No.28
Vol.2022-SLDM-198 No.28
Vol.2022-EMB-59 No.28
2022/3/10

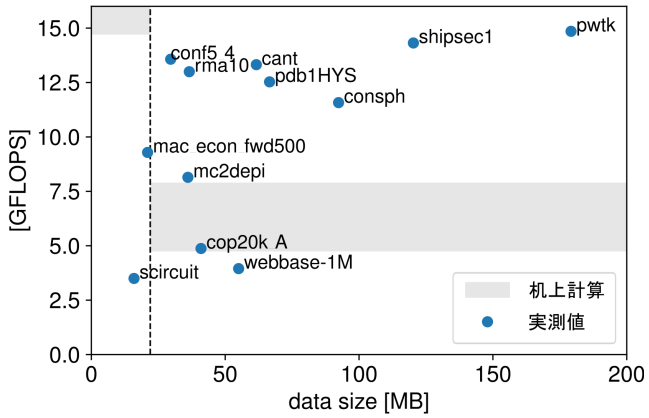

図 3　各疎行列のデータサイズと SpMV の実測性能

かった．そこで，PMNet を使った性能予測の利用を検討する．

PMNet ではワークロード実行時のメモリトレースを使ってキャッシュシミュレーションを行うため，どのキャッシュにどのデータがあるかを考慮して，性能推定するため，机上計算よりも精度良い推定が期待できる．

本章では，実測との比較により PMNet の精度検証を行い，本報告の目的であるメモリアーキテクチャ探索に利用できるかを検討する．ハードウェアパラメタや入力データによる性能の違いや優劣を知ることが重要であるため，推定値の絶対値が合う必要はないが相関が保たれている必要がある．そこで，入力データによる性能の違いをみるため，2.2 章と同じく HPCG および表 3 の疎行列を使った精度検証を行った．また，ハードウェアパラメタの影響をみるため，HPCG の疎行列を使った検証はいくつかのメインメモリのバンド幅について行った．

3.1　PMNet

PMNet [7] は，ワークロードを実際に実行した際のメモリトレース情報と実行命令数の情報を使って，任意のメモリアーキテクチャにおけるワークロード実行時間を推定するツールである．

メモリアーキテクチャは図 4 のようなグラフで定義する．アーキテクチャの定義に使用できるオブジェクトは，コア，キャッシュメモリ (L1/L2/L3)，メインメモリ，ルータである．コアは IPS，単精度浮動小数点演算性能，倍精度浮動小数点演算性能を，キャッシュメモリおよびメインメモリは，容量，ウェイ数，ブロックサイズ，読込および書込のバンド幅 (B/s) を設定できる．

PMNet では，まずグラフ上の各オブジェクトから全メモリオブジェクトへの最短経路を計算する．そしてメモリトレース 1 要素 (各コアが発行した Read または Write) 毎に，以下の動作を順次行う．

(1) アクセス対象アドレスがどのメモリオブジェクトにあるか特定する．

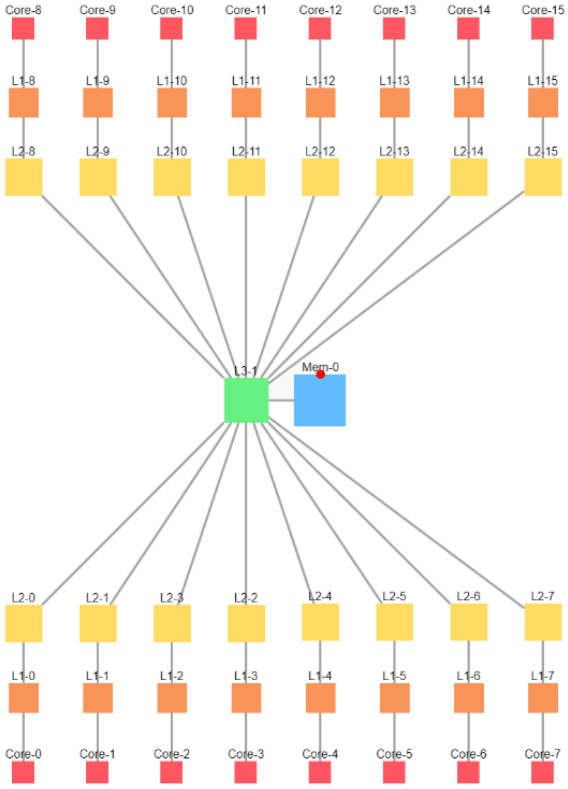

図 4　使用したメモリアーキテクチャ

(2) 予め計算した最短経路にしたがってアクセス情報をバケツリレーする．各キャッシュメモリオブジェクトは set associative シミュレータを持っており，キャッシュミスの時だけ次へ転送する．

(3) リレーの過程で，各オブジェクトのカウンタを通過データ量 (多くはブロックサイズ) だけ加算する．

終了後に，各オブジェクトの (積算データ通過量/バンド幅) = 占有時間 を計算し，占有時間の最大値を推定実行時間とする．このとき，最大占有時間をもつオブジェクトがボトルネックになっていると推定する．PMNet ではレイテンシを考慮していないことから，原則的に実測時間よりも推定時間は短く出る．

3.2　実験条件

PMNet に入力したメモリアーキテクチャのグラフを図 4 に示す．各要素のパラメタ設定は表 1, 2 に示した通りである．

ハードウェア変更による性能変化をみるため，3 つのメモリバンド幅で検証した．実測値測定においては，BIOS で 2 つ，4 つ，6 つの DIMM スロットのみを有効にすることで，メインメモリのバンド幅を変更した [13]．それぞれの状態で STREAM benchmark [8] により実効バンド幅を測定して得た値をメモリバンド幅として PMNet に入力した．

また，比較は 2.2 章と同様に実効性能を算出して行った．

Vol.2022-ARC-248 No.28
Vol.2022-SLDM-198 No.28
Vol.2022-EMB-59 No.28
2022/3/10

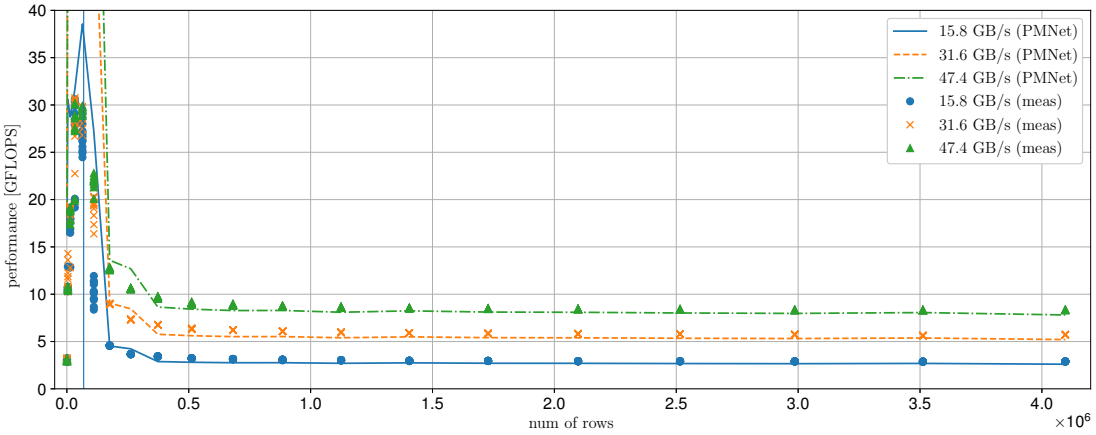

図 5　HPCG の疎行列を使った精度評価 (num of rows $< 4.2 \times 10^6$)

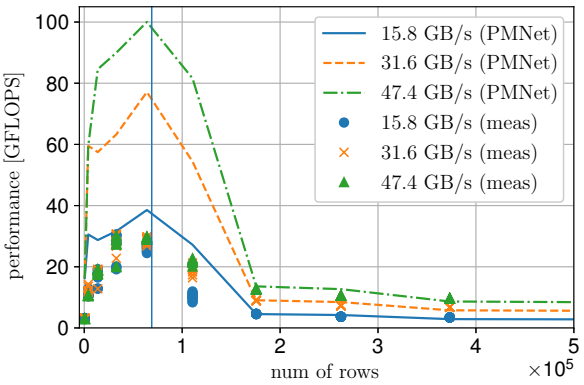

図 6　HPCG の疎行列を使った精度評価 (num of rows $< 5.0 \times 10^5$)

3.3　PMNet の精度評価

3.3.1　HPCG の疎行列を使った精度評価

図 5, 6 に PMNet による推定値と実際にワークロードを実行して測定した実測値を示す．2 図は同じデータを定義域を変えてプロットしたものである．実測は複数回行ったため，実測値のプロットは同じ行列サイズに複数の点が存在する．7.4×10^4 付近の縦線はデータサイズが L3 キャッシュ容量と等しくなる行列サイズを示している．2 章より，縦線の左側部分は演算ボトルネック，右側部分はメモリボトルネックになっていると考えられる．

PMNet のボトルネック推定結果によると，この線を境に演算ボトルネックからメモリボトルネックに変化していた．図からも，PMNet による推定値と実測値は，共にこの付近で性能値が下がっており，PMNet はボトルネック変化点を捉えることができていることがわかる．

演算ボトルネック時の性能について，PMNet による性能ピークの推定値は 38.6 GFLOPS, 77.1 GFLOPS, 100.0 GFLOPS だったのに対し，実測では 30.6 GFLOPS, 30.8 GFLOPS, 30.2 GFLOPS であり，最大で 3 倍以上の誤差となった (いずれもメモリバンド幅の小さい順)．机上計算値の誤差の原因となった，(1) 実行演算数の違い，(2) 実行 IPS の低下の内，前者については Intel SDE を使って取得した実行命令数を使うことで PMNet では考慮している．しかしながら，後者については推定速度低下の懸念から考慮に入れていない．このため，PMNet による推定値は机上計算による値よりは実測に近い値になるものの，誤差が残る結果となった．

メモリボトルネック時の性能については，行数 1.8×10^5 以降では実測と推定がよくあっていることがわかる．PMNet はキャッシュシミュレーションしているので，L3 キャッシュ容量がデータサイズよりも大きくなった直後の領域でも概ね正しく予測できている．ただし，ハードウェアプリフェッチは実装していない等，実際のキャッシュメモリの動きを完全にはシミュレートしていないため，PMNet と実測の間にも差が生じている．

3.3.2　SuiteSparce Matrix の疎行列を使った評価

図 7 に SuiteSparce Matrix の行列入力時の SpMV の性能の実測値と PMNet による推定値の比較を示す．

図では左側 5 つに PMNet によるボトルネック推定の結果が演算ボトルネックだったもの，右側 6 つにメモリボトルネックだったものを配置している． HPCG の疎行列の場合と同様，演算ボトルネックとなるワークロードでは推定値が実測値を大きく上回る結果となった．

メモリボトルネックだった行列について，2.2.2 章で述べた通り，机上計算での性能推定値は 4.74-7.90 GFLOPS になる．この範囲を斜線のバーで示したが，実測値の全てがこの範囲外になっている．PMNet ではキャッシュシミュレーションを使ってキャッシュヒット率も考慮しているため，机上計算で求めた上限よりも実測に近い推定ができている．PMNet の推定誤差を算出すると，$-75 \sim +42$ % と高精度とは言えないが，実測との相関係数は 0.80 と概ね傾向はつかむことができている．

Vol.2022-ARC-248 No.28
Vol.2022-SLDM-198 No.28
Vol.2022-EMB-59 No.28
2022/3/10

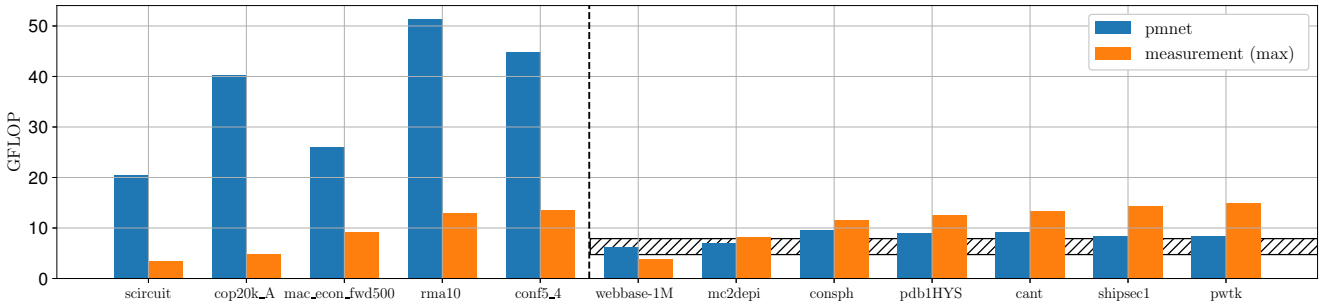

図 7　Suite Sparce Matrix を使った精度評価

以上から，PMNet は机上計算よりは実測に近い性能推定ができることがわかった．また，メモリボトルネック時の性能推定については，ハードウェア (メモリバンド幅) の変更を反映できており，アーキテクチャ検討に使用できると言える．ただしメモリバンド幅以外のパラメタ変更時の性能への影響については確認できておらず，今後検証が必要である．

4. PMNet によるアーキテクチャ検討

本章では，PMNet がターゲットとしている 2028 年に量産可能なプロセッサ関連のパラメタ値を IRDS ロードマップ [14] から抽出し，2028 年のプロセッサにおける SpMV の性能を予測する．その上で，SpMV の性能をより高めるために，ハードウェア構成の検討やソフトウェア最適化をどう進めるべきかについて提案する．

4.1 2028 年のプロセッサにおける疎行列ベクトル積実効性能

本章で使用した IRDS ロードマップ記載のパラメタを表 4 に示す．SIMD ポート数を 2 [15]，読込/書込時バンド幅は現行プロセッサと同じ，1 コア当たり

- L1: 128B/cycle, 64 B/cycle
- L2: 64B/cycle, 64 B/cycle
- L3: 64B/cycle, 64 B/cycle

とし，表 5, 6 のような設定で PMNet を実行した．メモリアーキテクチャは図 4 と同じものを使用した．ロードマップ上では 70 コアのところ，本稿では 16 コアのアーキテクチャで検討しているため，メインメモリのバンド幅はロードマップにある値に 16/70 を乗算した値を使用している．

PMNet を実行した結果を図 8 に示す (入力疎行列は HPCG)．PMNet のボトルネック推定の結果を参照すると，演算ボトルネックとなっているのは一番小さい行列 (8^3 行) のときだけで，16^3 行以上の行列では L2 キャッシュがボトルネックに，56^3 以上ではメモリがボトルネックになっていた．L2 キャッシュボトルネックとなる行列サイズが出てきたのは，演算性能が約 13.6 倍になったのに対し，キャッシュメモリのバンド幅は 1.7 倍しか増えていないためである．

表 4　IRDS ロードマップ [14] から引用したパラメタ

	2028
# cores per socket	70
Processor base frequency	3.9GHz
Core vector length	2048
L1 data cache size	44KB
L2 cache size	2.5 MB
LLC cache size	157 MB
HBM bandwidth	6.6 TB/s

表 5　2028 年の SpMV 性能予測に使用した Core の設定

倍精度浮動小数点演算性能	249.6 GFLOPS
時間当たり命令数 (IPS)	249.6 GIPS

表 6　2028 年の SpMV 性能予測に使用したキャッシュメモリおよびメインメモリの設定

	capacity	read bandwidth	write bandwidth
L1	44 KiB	473.6 GB/s	236.8 GB/s
L2	2.5 MiB	236.8 GB/s	236.8 GB/s
L3	2.25 MiB	236.8 GB/s	236.8 GB/s
Memory	-	1.51 TB/s	1.51 TB/s

L1/L2/L3 はコア当たり．Memory は CPU 当たり．

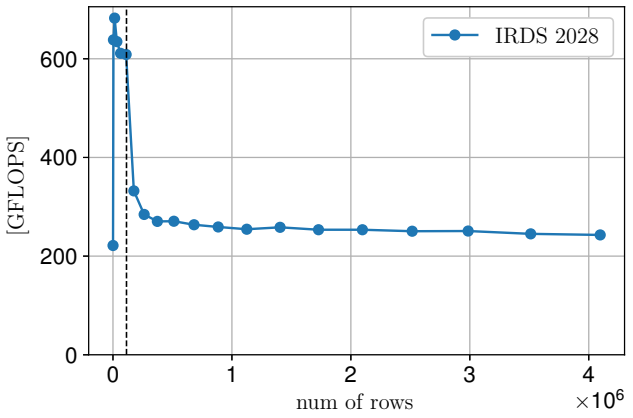

図 8　2028 年の SpMV 性能予測

Vol.2022-ARC-248 No.28
Vol.2022-SLDM-198 No.28
Vol.2022-EMB-59 No.28
2022/3/10

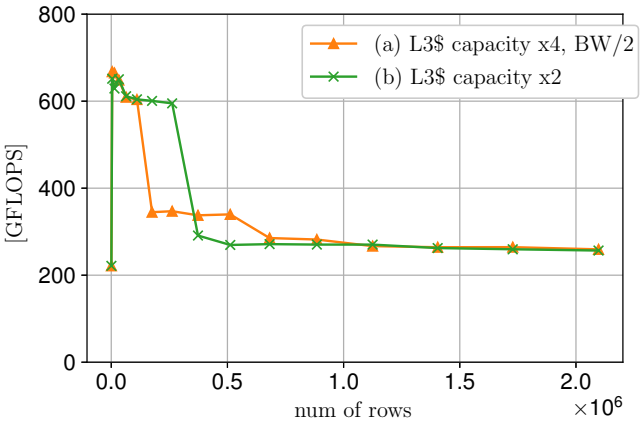

図 9　2028 年の SpMV 性能予測 - 3D キャッシュ想定例

4.2　ハードウェア構成検討

IRDS のロードマップには L3 キャッシュ容量が急激に増えている年はないため，3D キャッシュメモリは想定していないと考えられる．そこで，本章では，AMD 3D V-cache [16] のような三次元実装による L3 キャッシュ容量の増量を想定したハードウェアへの構成変更を検討する．

L3 キャッシュ容量が増えれば，メモリボトルネックとなる疎行列サイズを大きくすることができる．バンド幅を下げることなくキャッシュメモリ容量を大容量化することは困難であるが，3D キャッシュメモリであれば，容量を犠牲にし TSV (through silicon via) 数を増やすことでバンド幅低下を防ぐことができる．そこで，

(a) キャッシュ容量重視　L3 キャッシュ容量が 4 倍，L3 キャッシュのバンド幅が半分

(b) バンド幅重視　L3 キャッシュ容量が 2 倍，L3 キャッシュのバンド幅は 1 倍

の 2 つのシナリオを想定して，PMNet を実行した．結果を図 9 に示す．

(a) ではメモリボトルネックとなる演算の一部を改善できたものの，L3 キャッシュがボトルネックとなってしまったために，その改善量は 2 割程度と僅かとなった．一方 (b) では，改善できる演算は (a) より少ないものの，実効性能は 2 倍になった．

ハードウェア構成検討において，より幅広いワークロードの実効性能を改善できる (a) を選択するか，適用範囲が限定的でもより高い実効性能が得られる (b) を選択するかは，要求仕様次第だが，一般的にはより高い実効性能を得られる範囲の広い (b) が良いと考えられる．次章で述べるソフトウェア最適化の結果や他のワークロードの性能推定結果などを勘案して決めることになる．

4.3　ソフトウェア最適化指針の提案

図 9 で，(a) の性能が (b) の性能を上回っている範囲は限定的である．この範囲の疎行列演算について，アルゴリズム変更によるデータ配置の最適化ができれば，(b) が常に (a) を上回るようになる可能性がある．このようにハードウェア構成だけを変えて推定性能を比較するのではなく，それぞれの構成に合ったアルゴリズムを組み合わせた評価を行って比較することで，より適切なハードウェア構成を選択できる．

PMNet では各階層各キャッシュメモリのデータ処理時間 (占有時間) を推定できるので，どのキャッシュメモリに余裕があるのか，どのキャッシュメモリがボトルネックになっているかを瞬時に判定できる．この情報を元に，どのキャッシュメモリへデータを置くべきか検討し，アルゴリズム変更やソフトウェア・プリフェッチを実装することで，ソフトウェア最適化を行うことができる．ただし，現在の PMNet はプリフェッチには対応していないため，プリフェッチ効果を評価するには機能拡張が必要となる．

また，ハードウェア仕様決定後も，PMNet を使うことでハードウェア完成前に、ソフトウェアチューニングをすることができる．更に，キャッシュシミュレーション結果を参照し，キャッシュメモリの使用状況やミスヒット率を確認することで，性能チューニング支援も可能である．そのためには，キャッシュシミュレーションの精度向上が必要であり，場合によっては各種プリフェッチの他，キャッシュのプロトコルの実装も検討する必要がある．

5.　終わりに

本稿では，SpMV を例に PMNet の精度評価を行い，その実用性を検討した．その結果，PMNet は演算ボトルネック時の性能推定には適用が難しいものの，メモリボトルネックとなるワークロードについては，入力データおよびハードウェア (メモリバンド幅) による性能の違いを評価できることを示した．また，PMNet を使ったハードウェアやソフトウェアの最適化指針も示した．

今後は，PMNet の精度向上をするとともに，複数のワークロードを考慮した場合のメモリアーキテクチャ設計指針を検討する必要がある．また，本稿では 1CPU のみでの評価を行ったが，ハイパフォーマンスコンピューティング用途を考える場合，複数 CPU などより大規模な検討も必須となる．これについても今後行う予定である．

謝辞　PMNet は，国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の委託業務 (JPNP16007) の成果として得られたものである．

参考文献

[1] TOP500.org: TOP500 - November 2021, TOP500.org (online), available from ⟨https://www.top500.org/lists/top500/2021/11/⟩ (accessed 2022-01-07).
[2] TOP500.org: HPCG - November 2021, TOP500.org (online), available from ⟨https://www.top500.org/lists/hpcg/2021/11/⟩ (accessed 2022-01-07).

[3] I. R. eguly and M. Giles: Efficient sparse matrix-vector multiplication on cache-based GPUs, *2012 Innovative Parallel Computing (InPar)*, pp. 1–12 (2012).
[4] Z. Zhang, H. Wang, S. Han and W. J. Dally: SpArch: Efficient Architecture for Sparse Matrix Multiplication, *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 261–274 (2020).
[5] I. B. Peng and J. S. Vetter: Siena: Exploring the Design Space of Heterogeneous Memory Systems, *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 427–440 (2018).
[6] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee and V. Srinivas: CACTI 7: New Tools for Interconnect Exploration in Innovative Off-Chip Memories, *ACM Transactions on Architecture and Code Optimization*, Vol. 14, No. 2 (2017).
[7] 幸 朋矢，遠藤敏夫：次世代高性能計算ノードにむけたメモリアーキテクチャ探索のためのツールチェーン，情報処理学会研究報告，Vol. 2022-ARC-240 (2022).
[8] J. D. McCalpin: Memory Bandwidth and Machine Balance in Current High Performance Computers, *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25 (1995).
[9] Intel: Intel Xeon Gold 5218 Processor 22M Cache 2.30 GHz Product Specifications, Intel (online), available from ⟨https://ark.intel.com/content/www/us/en/ark/products/192444/intel-xeon-gold-5218-processor-22m-cache-2-30-ghz.html⟩ (accessed 2022-01-17).
[10] WikiChip: Cascade Lake - Microarchitectures - Intel, WikiChip (online), available from ⟨https://en.wikichip.org/wiki/intel/microarchitectures/cascade_lake⟩ (accessed 2022-01-25).
[11] T. A. Davis and Y. Hu: The University of Florida Sparse Matrix Collection, *ACM Transactions on Mathematical Software*, Vol. 38, No. 1 (2011).
[12] S. Williams, L. Oliker, R. Vuduc, J. Shalf, K. Yelick and J. Demmel: Optimization of sparse matrix-vector multiplication on emerging multicore platforms, *SC '07: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*, pp. 1–12 (2007).
[13] FUJITSU: ホワイトペーパー FUJITSU Server PRIMERGY & PRIMEQUEST Xeon スケーラブル・プロセッサ (Cascade Lake-SP) 搭載システムのメモリパフォーマンス (2017).
[14] IRDS: International Roadmap for Devices and Systems (TM) 2021 Edition, IEEE (online), available from ⟨https://irds.ieee.org/editions/2021⟩ (accessed 2022-01-07).
[15] 近藤正章ほか：White Paper 1.0.0, NGACI (online), available from ⟨https://sites.google.com/view/ngaci/home⟩ (accessed 2022-01-11).
[16] AMD: 3D V-Cache, AMD (online), available from ⟨https://www.amd.com/en/campaigns/3d-v-cache⟩ (accessed 2022-01-25).