

# 並列処理による計算の高速化

システム科学技術学部 電子情報システム学科

1年 土門拓実

1年 由利廉太郎

1年 渡辺健太

指導教員 システム科学技術学部 電子情報システム学科

助教 中村真輔

准教授 廣田千明

## 1.背景と目的

近年の CPU はメニーコア化が進んでいる。その背景として単一コアの CPU ではクロックの数値が計算時間に直結するがその限界が見えてきた。そのため 1 つの CPU に複数のコアを搭載したマルチコア CPU が登場した。それに伴いプログラム側で計算の高速化の方法として並列計算が用いられるようになった。並列計算とはプログラムを複数のスレッドで分担して同時に処理することで計算の高速化をはかるという手法である。

Ryzen シリーズは 2017 年に発売された AMD 製の CPU である。AMD は以前 Intel と競争関係にあったが長らくシェアで Intel に負けていた。そこでこれまでとはまったくちがう Zen アーキテクチャを用いた CPU を開発した。これが Ryzen シリーズである。今回使用した Ryzen7 1700 は「効率的で強力なマルチコア処理」 [3]を謳うように、発売当時コンシューマー向け CPU としては最多の 8 コア 16 スレッドを実現した CPU であった。これは Ryzen7 が対抗馬として挙げている Intel の Core i7 4770K のコア数、スレッド数の 2 倍である。しかしながら価格はほぼ同じである。このように型番や宣伝方法、価格などから Intel の Core i シリーズを強く意識していることがわかる。AMD 製の CPU は Intel 製の CPU に比べ安価であることが多いため Ryzen の有用性が証明できればメニーコアの計算環境を安く構築することができるのではないかと考えられる。

今回の研究の目的は AMD 製のコンシューマー向け CPU である Ryzen7 と Intel 製の CPU である Xeon を比較することで Ryzen7 の計算用途での性能を調査するというものである。

## 2.並列計算について

### 2.1.スレッド並列

スレッド並列とは複数のスレッドを用いて、計算を分割し、同時進行することによって計算時間を短縮する方法である。マルチコア CPU と並列化に対応したプログラムを組み合わせることでスレッド並列が実現できる。

### 2.2 .OpenMP

OpenMP は並列プログラムの手法の一つである。少しの指示文を追加するだけで、普通のプログラムを並列化されたプログラムへ変換でき、コンパイラが OpenMP 対応であれば、どのような環境でも使用できる[1]。また、異なる OS 環境で同じプログラミング手法が利用できるプログラミング手法である[2]。

指示文の形式は「`#pragma omp 指示文名 [ 指示句 [ [, ] 指示句 ] ... ]`」であり、処理ブロックの前に指定することで並列化することができる。以下のようなプログラムが例として挙げられる。

```
#pragma omp parallel for num_threads(4)
for(i=0; i<n; i++){
    printf("i=%d",i);
}
```

ここで用いた指示文は `parallel for` である。 `parallel for` は `for` 文を並列化する際に用いる指示文である。また、用いた指示句は `num_threads` である。 `num_threads` は使用するスレッドの数を指定することができる指示句である。

### 2.3.性能評価のための指標

並列計算機の性能を測る指標としては、スピードアップ、効率、FLOPS の 3 つが頻繁に用いられる。スピードアップは並列化する前と後で速度が何倍になったかを示す指標である。効率はプロセッサ数を増やすことによってどの程度のスピードアップが得られたかを示す指標である。FLOPS は 1 秒間に実数演算が何回できたかを示す指標である。それぞれの導出方法を以下に示す。

$$(\text{スピードアップ}) = \frac{(\text{逐次での時間})}{(\text{並列での時間})} \quad (\text{効率}) = \frac{(\text{スピードアップ})}{(\text{プロセッサ数})} \quad (\text{FLOPS}) = \frac{(\text{実数演算の回数})}{(\text{計算時間})}$$

## 3.実験内容と結果

### 3.1.計算内容

今回行う計算は行列と行列の積である。用いる行列はすべて正方行列とし、行・列数はそれぞれ 1000, 2000, 4000 と 2 倍ずつ増やしていく。逐次処理の時計算時間が 2 時間を超えたところを限界とする。

### 3.2.計算環境

以下の 2 つの環境についてそれぞれプログラムを実行し、2.3 の指標のうち効率と FLOPS について評価を行う。

①CPU:Ryzen 7 1700 3.0GHz 8 コア 16 スレッド [3]

メモリ : 16GB

コンパイラ : gcc 4.8.5

②CPU:Xeon E5-2695v2 2.4GHz 12 コア 24 スレッド ×2 [4]

メモリ:128GB

コンパイラ : icc 16.0.3

### 3.3.実験結果

3.2に示したそれぞれの環境での2.3の指標のうち効率とFLOPSについての結果をそれぞれ示す.

表1 ①の効率

行・列数 \ スレッド数	1000	2000	4000	8000
4	0.872	0.973	0.864	0.875
8	0.852	1.054	0.850	0.864

表2 ①のFLOPS(GFLOPS)

行・列数 \ スレッド数	1000	2000	4000	8000
1	0.449	0.297	0.410	0.392
4	1.565	1.155	1.417	1.373
8	3.059	2.501	2.788	2.711

表3 ②の効率

行・列数 \ スレッド数	1000	2000	4000	8000	16000
4	0.921	0.872	0.916	0.866	0.877
12	0.757	0.633	0.657	0.621	0.635
24	0.494	0.884	0.869	0.779	0.773

表4 ②のFLOPS(GFLOPS)

行・列数 \ スレッド数	1000	2000	4000	8000	16000
1	3.280	1.982	1.505	1.591	1.586
4	14.073	6.912	5.513	5.509	5.561
12	34.682	15.053	11.866	11.863	12.088
24	45.283	42.057	31.405	29.767	29.428

### 4.考察

逐次処理での時間について Ryzen の定格クロックの方が大きいにもかかわらず結果は Xeon の方が早かった. これは環境の都合上異なるコンパイラを使ったことによるものと思われる.

効率に着目すると8スレッド以上の処理での効率が Ryzen の方が優れているといえる.

Xeon の 24 スレッドでのスピードアップと効率に着目すると 2000 行 2000 列の計算以降の値が大きく上昇している。これは 12 コアの Xeon を 2 機使用する状況のためキャッシュも単純計算で 2 倍となり、より高速な計算が行えたことによるものと思われる。ここで行列のデータサイズとキャッシュ容量を比較してみると以下のようになった。この時行列のデータサイズは次の式で求めたものとする。

$$\text{行列のデータサイズ} = (\text{行} \cdot \text{列数})^2 \times (\text{行列の数}) \times 8 \text{ バイト}$$

表 5 ②の行列のデータサイズとキャッシュ容量の比較

1000 行 1000 列のデータサイズ	15MB
2000 行 2000 列のデータサイズ	61MB
CPU1 つあたりのキャッシュ容量	30MB
CPU2 つでのキャッシュ容量	60MB

2000 行 2000 列の行列ではデータのほとんどが 24 スレッドでのキャッシュ容量に入っている。そのためスピードアップ、効率ともに高いスコアを出すことができたと思われる。それに対し、1000 行 1000 列のデータサイズは CPU1 つのキャッシュにおさまってしまう。2 つの CPU 間のキャッシュは共有されていないため、データのやり取りに時間がかかり 2000 行 2000 列の行列を計算した時に比べスコアが伸びなかった可能性があげられる。

## 5.まとめ

Ryzen と Xeon を比較した結果 Ryzen のほうが優れている点として効率のよさがあげられる。しかし、計算環境を総合的に判断すると Intel 製の計算環境の方が高速な計算を可能にするといえるだろう。今回はコンシューマー用 CPU とサーバー用 CPU を比較したが用途が異なるため比較対象として適切でなかったかもしれない。また、ソフトウェアとハードウェアの相性によっても結果が変わってくると思われるので計算環境を作るうえでそれらの相性を考えることも重要になるだろう。

## 参考文献

- [1] 「OpenMP 入門」北山洋幸著 秀和システム
- [2] 「OpenMP 並列プログラミング」菅原清文著 カットシステム
- [3] 「AMD Ryzen™ 7 1700 プロセッサー」  
<https://www.amd.com/ja/products/cpu/amd-ryzen-7-1700> (2018/3/5)
- [4] 「インテル® Xeon® プロセッサー E5-2695 v2」  
<https://ark.intel.com/ja/products/75281/Intel-Xeon-Processor-E5-2695-v2-30M-Cache-2-40-GHz> (2018/3/5)