

ラズパイ 並列プログラミング入門

山際 伸一

リスト1 行列の各要素の平均値を求めるプログラム(逐次処理)

```
#define N 1000
#define M 1000

int main(){
    int i, j;
    double a[N][M];
    double sum;

    sum = 0.0;
    for (i=0; i<N; i++)
        for (j=0; j<M; j++)
            sum += a[i][j];

    sum /= N*M;
}
```

ここでは行列の各要素の平均値を求めるプログラムを考えながら、並列プログラミングについて説明します。第7章で説明した環境を構築している方は、ぜひ試してみてください。

単純に、行列の各要素の平均値を計算するプログラムリスト1に示します。

自力で並列化は簡単じゃない

共有メモリ方式のプログラムを実装しようと思うと、例えば以下の方法があります。

- 複数のプロセスを立ち上げ、ファイルを通じて共有するデータを交換する
- 複数のスレッドをプロセス内で起動し、同じ変数を介して共有するデータを交換する

しかし、これらのプログラムの構成を考えるととても複雑です。

● 行列を複数のスレッドで共有しながら並列計算する

行列aを複数のスレッドで共有しながら、並列に計算するようにこのプログラムを変形してみます。例えば図1のように、行列をスレッド数分のブロックに分け、そのブロック内の合計値を求め、もう一つの共有している変数であるsumに担当する部分を書き込んで、最後にsumの全ての要素の合計と平均を計算します。

スレッドを使って並列化した場合、リスト2のような構造になります。

N_THREADにスレッド数を指定します。

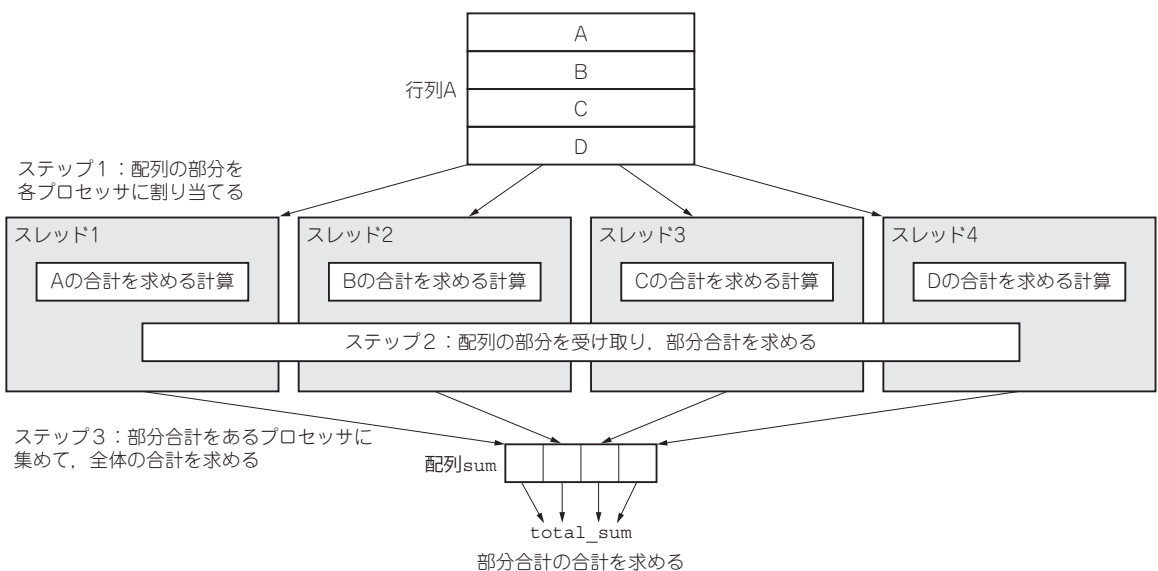


図1 行列をスレッド数分のブロックに分けて計算する