3.2.2. 並列効果の低下とラージページの関係について

富士通株式会社　　内藤 俊也

この報告は、C，C++，Fortran の速度比較において、C 言語の OpenMP の並列効果でみられた性能低下の現象について調査結果を報告するものです。

1. はじめに

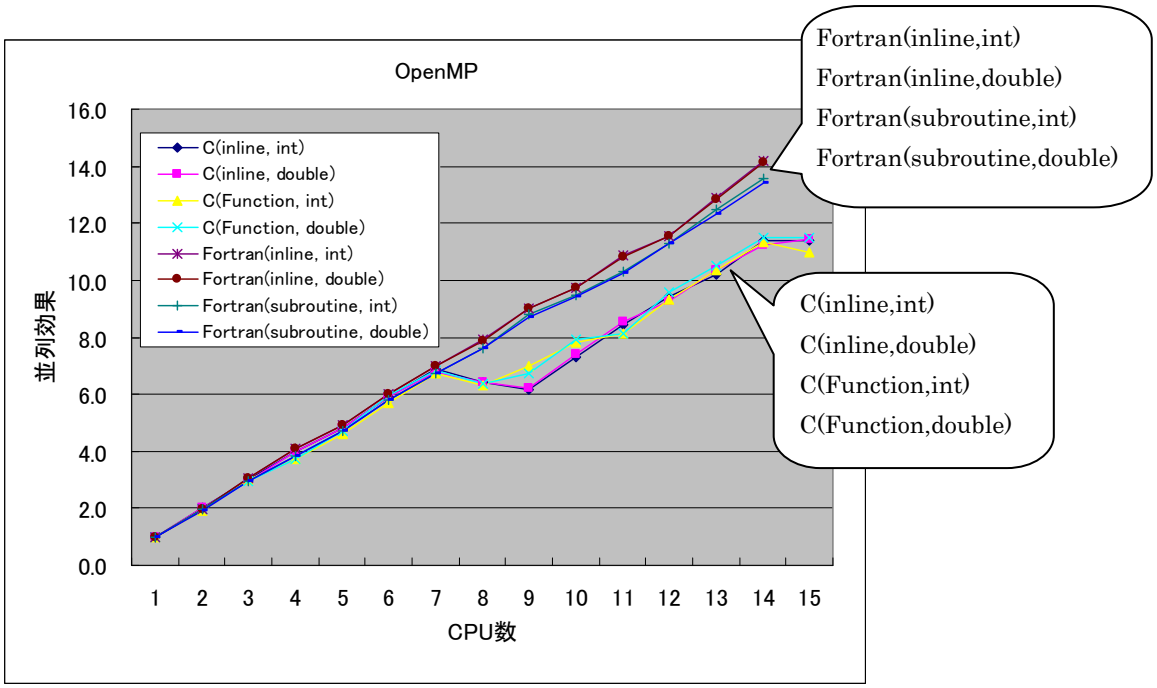
C OpenMP 性能が 8 並列以上で並列効果（※1）が低下していることについて、原因を説明します。まず、使用したプログラムの一部を下記に示します。

プログラムの一部（C 言語バージョン）

```
/* Median filter */
for( z=mask_size/2; z<image_size-mask_size/2; z++ ){      /* このループを並列化 */
  for( y=mask_size/2; y<image_size-mask_size/2; y++ ){
    for( x=mask_size/2; x<image_size-mask_size/2; x++ ){
      i = 0;
      for( zz=-mask_size/2; zz<=mask_size/2; zz++ ){
        for( yy=-mask_size/2; yy<=mask_size/2; yy++ ){
          for( xx=-mask_size/2; xx<=mask_size/2; xx++ ){
            list[i] = image[(x+xx)+(y+yy)*image_size+(z+zz)*image_size*image_size];
            i++;
          }
        }
      }
      /* Bubble sort */
      for( i=0; i<mask_size*mask_size*mask_size-1; i++ ){
        for( j=1; j<mask_size*mask_size*mask_size; j++ ){
          if( list[i]<list[j] ){
            tmp = list[i]; list[i] = list[j]; list[j] = tmp;
          }
        }
      }
        median[x+y*image_size+z*image_size*image_size] = list[mask_size*mask_size*mask_size/2];
    }
  }
}
```

このプログラムは 3 次元画像処理で画像中のスパイクノイズを除去するために比較的頻繁に使われるものです。下記のグラフで、subroutine あるいは Function と表記されたものはメディアンフィルタを関数として実装したもの、inline と表記されたものは関数化しなかったものを表します。また、int あるいは double は配列 image が整数型であるか倍精度浮動小数点型であるかを表します。

以下のグラフと測定値で、4 つの OpenMP　C プログラム（インライン | 関数、整数 | 浮動小数点）は、ほぼ同じ現象であるため、「インライン、整数」を取り上げて説明します。



計算時間（秒）		C				Fortran			
		インライン		関数		インライン		subroutine	
	CPU数	整数	浮動小数点	整数	浮動小数点	整数	浮動小数点	整数	浮動小数点
OpenMP	1	6.490	6.740	7.260	7.470	6.460	6.811	6.218	6.607
	2	3.220	3.370	3.810	3.770	3.245	3.419	3.146	3.410
	3	2.177	2.280	2.433	2.510	2.118	2.227	2.123	2.253
	4	1.620	1.690	1.940	1.990	1.577	1.668	1.630	1.736
	5	1.350	1.402	1.570	1.592	1.315	1.383	1.322	1.406
	6	1.100	1.140	1.272	1.250	1.076	1.134	1.071	1.142
	7	0.941	0.990	1.080	1.090	0.925	0.975	0.921	0.980
	8	1.010	1.050	1.150	1.170	0.818	0.863	0.816	0.870
	9	1.050	1.084	1.040	1.108	0.716	0.754	0.706	0.759
	10	0.890	0.910	0.931	0.941	0.664	0.700	0.655	0.702
	11	0.769	0.789	0.891	0.918	0.595	0.628	0.602	0.646
	12	0.690	0.727	0.780	0.780	0.560	0.590	0.550	0.584
	13	0.635	0.650	0.700	0.711	0.502	0.531	0.498	0.536
	14	0.571	0.600	0.640	0.651	0.456	0.481	0.458	0.493
	15	0.570	0.590	0.660	0.650	–	–	0.452	0.483

※1：並列効果は、clock( )でループの実行時間を計測した CPU 時間を、並列数で割り 1 並列を基準時間（=1.0）として表現しています。ループの時間計測は、omp parallel for 構文の直前と直後で clock( ) を呼び出して区間計測しています。
なお、clock(3C)は、プロセス（全スレッドの合計）の usr+sys 時間を計測します。

2. 性能低下の原因について

① C OpenMP　8 並列以上での性能低下の原因

1～7 並列と比較して、8 並列以上の場合に新たなラージページセグメントを獲得するコスト（約 0.15 秒～0.2 秒）が発生しており、そのため、性能低下に見えています。これを以下に説明します。

ラージページセグメントの獲得量を NQS (qsub –oi)の統計情報から採取しました。

下の表は、並列数毎に個別のジョブ（計 15 個のジョブ）として実行し、並列数毎のラージページ使用量を計測したものです。「Used Resource」の「Max Large Page」が獲得したラージページセグメントの合計を表しており、8 並列以上の場合に新たなラージページセグメントの獲得が発生しています。

並列数	ラージページ		
	Allocated Resource (MB)	Used Resource Average (MB)	Used Resource Max(MB)
1	3072	974	1024
2	3072	935	1024
3	3072	947	1024
4	3072	811	1024
5	3072	774	1024
6	3072	807	1024
7	3072	861	1024
8	3072	1425	2048
9	3072	1457	2048
10	3072	1324	2048
11	3072	1343	2048
12	3072	1263	2048
13	3072	1208	2048
14	3072	1205	2048
15	3072	1144	2048

8 並列以上で、新たなラージページセグメントの獲得が発生

測定したマシンの設定では、スレッド毎のスタック領域（**※2**）が 128MB になっているため、プログラム全体で必要なスタックサイズは、128MB×並列数となります。プログラムが獲得するスタックサイズは、8 並列のときに 1024MB になり、ヒープ域を合わせたラージページ使用量が初期獲得量（1GB）を超えることが計算上でも分かります。このため、8 並列以上の場合に、新たなラージページセグメント獲得のためのコストが発生しています。

新たなラージページセグメント獲得のためのコストは、約 0.15 秒 ～ 0.2 秒であり、実行時間が長ければ無視できるほどのコストであると考えます。今回の 8 並列測定結果は約 1 秒であるため、コストの割合が多く見えていて、グラフではこのコストが顕著に表れています。

今回の報告は OpenMP の調査結果ですが、**自動並列時も同じ現象**になります。

また、測定するマシンの設定により、新たなラージページセグメント獲得の並列数が変わりますので注意して下さい。

※2：スレッド毎のスタック領域は、基本的にプロセスのスタック領域と同じ大きさです。

② C と Fortran の違い(何故 Fortran で発生しないか)

ラージページセグメントはスレッド生成時に獲得されますが、スレッド生成のタイミングが C と Fortran とで異なる場合があります。

C の場合、常に omp parallel ディレクティブが現れた位置にスレッドが生成されます。

Fortran の場合、主プログラム内に OpenMP ディレクティブが存在する場合に特殊な処理を行っています。この処理は主プログラムの先頭にダミーの omp parallel があるかのようなオブジェクトプログラムを生成しますが、このダミーの omp parallel の時点でスレッドが生成され、ラージページセグメントを獲得するコストが Fortran の場合に区間計測の外側で行われることになります。これが C と Fortran の違いになります。

しかし、プログラム全体の計測時間であれば、ラージページセグメントを獲得するコストが C と Fortran 共に含まれるため、**C と Fortran と同様の現象**になります。

3. 補足

本件の報告とは直接関係ありませんが、グラフと測定値より 14 並列と 15 並列で並列効果が変わらないように見えます。これは並列化されるループの回転数が 126 回転であり、14 並列では各スレッド 9 回転、15 並列でも最大 9 回転（8 回転と 9 回転のスレッドが混在）となるためです。

以上