2.3. アプリ性能

2.3.1. Intel クアッドコア CPU でのベンチマーク

東京海洋大学　　吉岡　諭

1. はじめに

この数年でマルチコア CPU の普及が進んできた。x86 系の CPU でも Intel と AMD がデュアルコア、クアッドコアの CPU を次々と市場に送り出していて、それらが PC クラスタの CPU として採用され、HPC に活用されている。ここでは Intel クアッドコア CPU を搭載した PC 単体で浮動小数点を中心としたベンチマークプログラムを実行し、CPU 及びコンパイラの並列性能を調べた結果を報告する。並列化には MPI、OpenMP、コンパイラの自動並列化機能を用いた。

2. ベンチマーク

今回ベンチマークプログラムとしては姫野ベンチマークを採用した。姫野ベンチマークがマルチコア CPU の性能を測定する最良のベンチマークということでは必ずしもないが、これまで日本で浮動小数点演算の実効性能を測定するベンチマークとして広く利用されてきたことを考慮し採用した。姫野ベンチマークは 3 次元格子でポアッソン方程式をヤコビの反復法で解く場合に主要となるループの処理速度を計るものである。その主要ループは以下の通りである。

```
gosa= 0.0
do k=2,kmax-1
   do j=2,jmax-1
      do i=2,imax-1
         s0=a(i,j,k,1)*p(i+1,j,k) &
              +a(i,j,k,2)*p(i,j+1,k) &
              +a(i,j,k,3)*p(i,j,k+1) &
              +b(i,j,k,1)*(p(i+1,j+1,k)-p(i+1,j-1,k) &
                          -p(i-1,j+1,k)+p(i-1,j-1,k)) &
              +b(i,j,k,2)*(p(i,j+1,k+1)-p(i,j-1,k+1) &
                          -p(i,j+1,k-1)+p(i,j-1,k-1)) &
              +b(i,j,k,3)*(p(i+1,j,k+1)-p(i-1,j,k+1) &
                          -p(i+1,j,k-1)+p(i-1,j,k-1)) &
              +c(i,j,k,1)*p(i-1,j,k) &
              +c(i,j,k,2)*p(i,j-1,k) &
              +c(i,j,k,3)*p(i,j,k-1)+wrk1(i,j,k)
         ss=(s0*a(i,j,k,4)-p(i,j,k))*bnd(i,j,k)
         gosa=gosa+ss*ss
         wrk2(i,j,k)=p(i,j,k)+omega *ss
      enddo
   enddo
```

測定に用いたプラットホームは以下のものである。

(1) Intel Core 2 Extreme QX6700 (4 コア)

model name :　Intel(R) Core2 Quad CPU　(Kentsfield) 2.66GHz
L2 cache size :　4096 KB×2
FSB :　1066MHz
OS :　CentOS 5.0 for Intel64

(2) Intel Xeon　E5462　2CPU (4x2 コア)

model name :　Intel(R) Xeon E5462 Quad CPU　(Harpertown)　2.80GHz
L2 cache size :　6MB× 2/cpu
FSB :　1600MHz
OS :　Fedora 8 for Intel64

(3) Intel Core i7 940　(4 コア)

model name :　Intel(R) Core i7 Quad CPU　(Nehalem)　2.93GHz
L2 cache size :　256kB/core
L3 cache size :　8MB (共有)
QPI :　4.8GHz
OS :　Cent OS 5.2 for Intel64

なお、STREAM ベンチマークを用いて測定した両システムのメモリバンド幅は以下の通りである。

(1) QX6700 :1 コアでは 4.6GB/s〜4.7GB/s、4 コア(openmp)でも 4.6GB/s〜4.7GB/s
(2) E5462 :1 コアでは 4.7GB/s〜5.5GB/s、8 コア(openmp)では 8.1GB/s〜8.8GB/s
(3) Core i7 940:1 コアでは 5.7GB/s〜10GB/s、4 コア(openmp)では 11GB/s〜16GB/s

測定に用いたコンパイラは以下のものである。

A) Fujitsu Fortran Version 3.0
B) Intel Fortran Compliler 10.0.026 (Core i7 940 だけ 11.1.056)
C) GNU gfortran 4.1.2

3. 測定結果

ベンチマークの測定は 4 つの格子サイズで行った。

XS (64x32x32)
S (128x64x64)
M (256x128x128)
L (512x256x256)

利用する配列の総バイト数は、XS では 3.6MB、S では 29MB、M では 235MB、L では 1.9GB になる。
測定結果を以下に示す。

3.1 スカラー性能

コンパイラオプションは以下の通りである。

gfortran -O3 (GNU)、ifort -O3 (Intel)、 frt -Kfast (Fujitsu)

測定結果を図 1 に示す。どのシステムとも格子サイズが大きくなるに従って、性能が落ちている。特に XS と S の間で性能差が大きい。XS では全配列が L2 キャッシュにおさまるのに対して、S 以上の格子サイズでは配列が L2 キャッシュから溢れていることが原因と考えられる。また、Core i7 940 (Nehalem)のスカラー性能が高いことが目立つ。

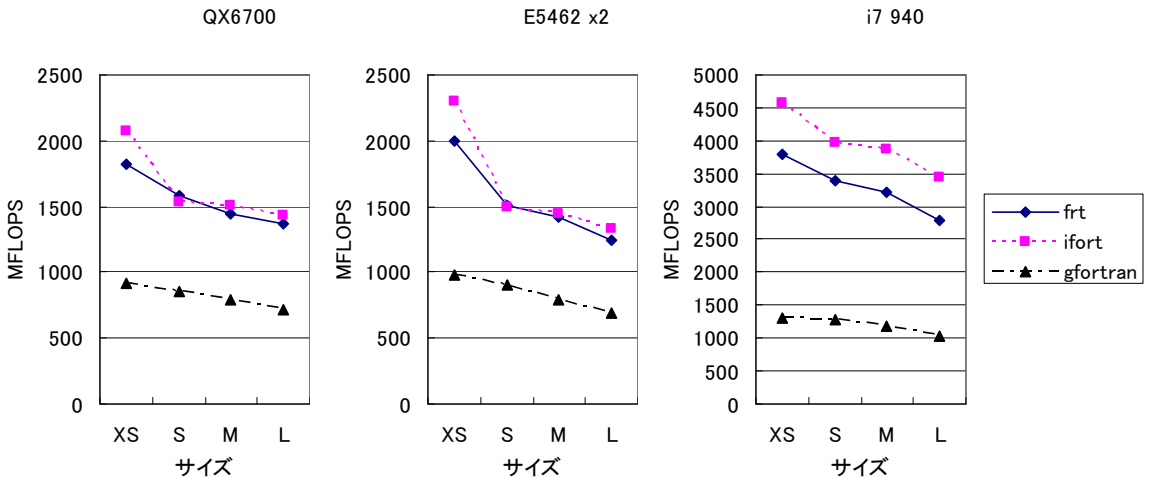

図 1. スカラー性能

3.2 自動並列化

GNU gfortran は自動並列化機能を持っていない。

スカラー版のベンチマークコードはそのままでは Intel コンパイラ ver10.0 及び ver11.1 では自動並列化されなかった。そのため、わずかなコードの変更とコンパイルオプションの追加を行うことによって、自動並列化を行った。

コンパイラオプションは以下の通りである。

ifort -O3 -parallel -par-threshold99 (Intel)、 frt -Kfast,parallel (Fujitsu)

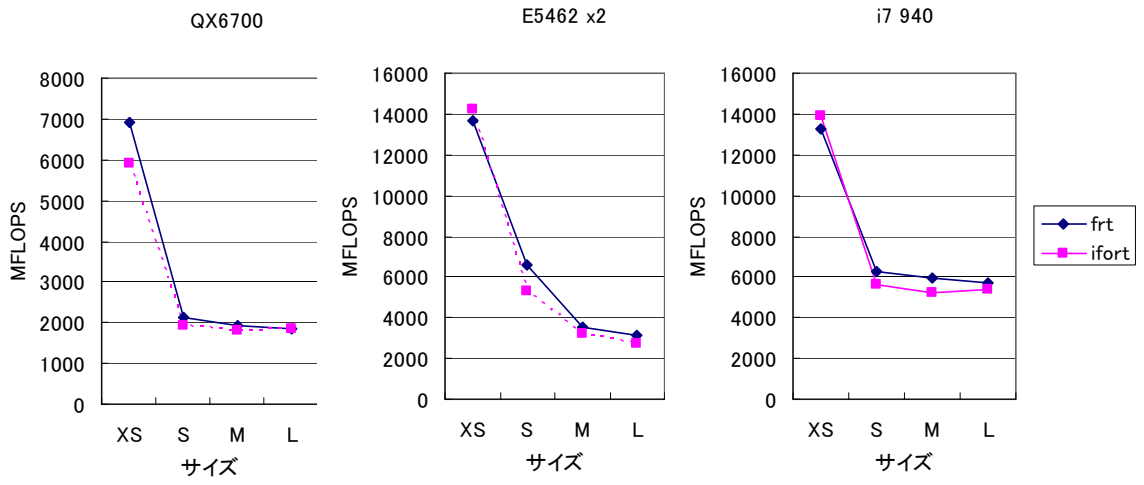

図 2. 自動並列化性能

測定結果を図 2 に示す。ここでも各システムとも格子サイズが大きくなるに従って、性能が落ちている。スカラー計算に対する性能向上率は富士通コンパイラでは、QX6700 で 1.35 倍(L サイズ)から 3.78 倍(XS サイズ)、E5462 で 2.52 倍(L サイズ)から 6.85 倍(XS サイズ)、i7 940 で 2.07 倍(L サイズ)から 3.51 倍(XS サイズ)となっている。格子サイズ XS では性能向上率がかなり高く、L2 キャッシュにデータがおさまっている場合には、マルチコアの威力が発揮できているようである。 Core i7 940 はコア数が 4 つであるのに、その倍のコア数の E5462x2 と同等以上の性能を示している。

3.3 OpenMP

コンパイラオプションは以下の通りである。

gfortran -O3 -fopenmp (GNU)、ifort -O3 -openmp (Intel)、 frt -Kfast,OMP (Fujitsu)

測定結果を図 3 に示す。ここでも各システムとも格子サイズが大きくなるに従って、性能が落ちている。GNU コンパイラの OpenMP 性能はかなり低いこともわかる。スカラー計算に対する性能向上率は富士通コンパイラでは、QX6700 で 1.36 倍(L サイズ)から 3.85 倍(XS サイズ)、E5462 で 2.40 倍(L サイズ)から 6.66 倍(XS サイズ)、i7 940 で 2.06 倍(L サイズ)から 3.50 倍(XS サイズ)となっている。自動並列化と同様に、格子サイズ XS では性能向上率がかなり高い。自動並列の場合と同様に Core i7 940 は E5462x2 と同等以上の性能を示している。

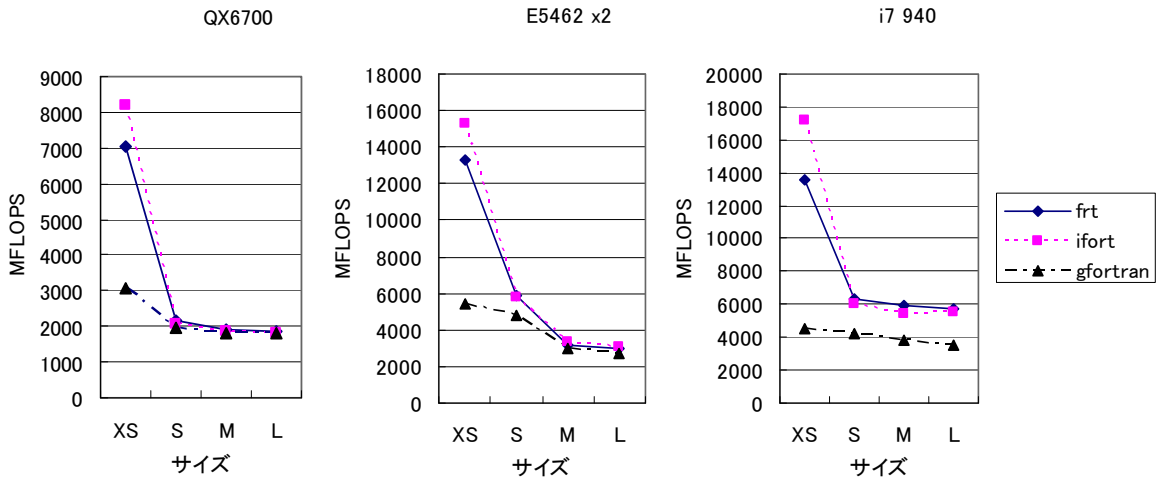

図 3. OpenMP 性能

3.4 MPI

MPI ライブラリは OpenMPI 1.2.4 を用いた。1 次元方向の領域分割を用いて並列化をしている。

コンパイラオプションは以下の通りである。

gfortran -O3 (GNU)、ifort -O3 (Intel)、 frt -Kfast (Fujitsu)

測定結果を図 4 に示す。各システムとも格子サイズが大ききなるに従って、性能が落ちている。スカラー計算に対する性能向上率は富士通コンパイラでは、QX6700 で 1.33 倍(L サイズ)から 2.79 倍(XS サイズ)、E5462 で 2.00 倍(L サイズ)から 6.42 倍(XS サイズ)、i7 940 で 2.23 倍(L サイズ)から 3.51 倍(XS サイズ)となっている。性能向上率は他の並列計算に比べてわずかに小さい。

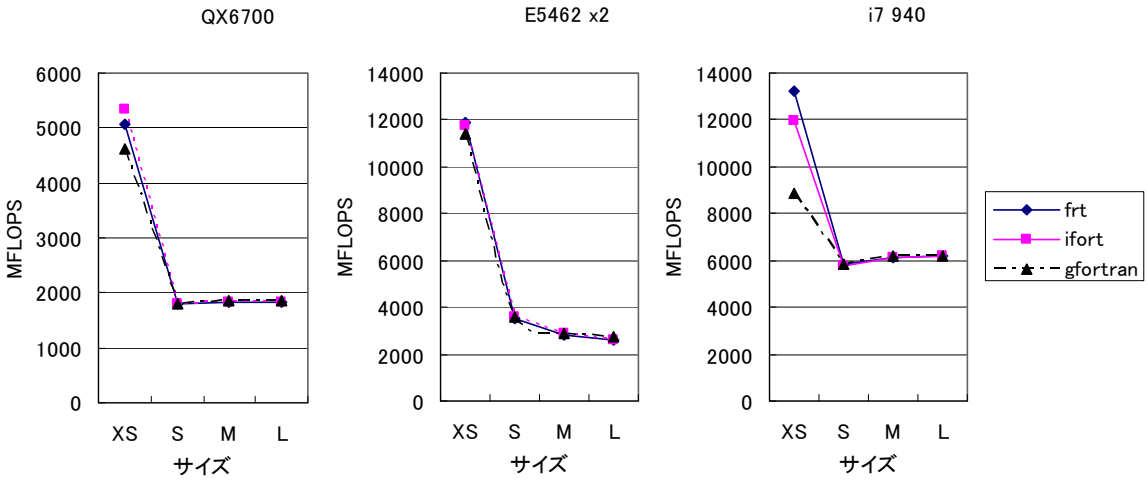

図 4. MPI 性能

なお、すべての並列計算で、QX6700 では S～L サイズで同等の結果なのに対して、E5462 では S サイズの性能が M、L サイズの性能を大きく上回っている。その理由として、利用できる L2 キャッシュの総量が QX6700 では 8MB なのに対して、E5462 では 24MB あり、これは S サイズの計算が必要とするメモリ量にほぼ匹敵し、ある程度 L2 キャッシュを利用した計算ができているためと考えられる。

Core i7 940 は L サイズの計算で E5462x2 の倍の性能を示している。

3.5 コア数と並列性能

E5462 で富士通コンパイラを用いて、利用するコア数(スレッド数)に対する性能測定を自動並列、OpenMP、

MPI それぞれに対して測定した結果を図 5 に示す。基本的には利用するコア数が増すにつれて性能は上がっているが、MPI では 8 コアの性能が 4 コアの性能を下回る場合がある。

4. 考察

4.1 コンパイラ

スカラ性能は富士通コンパイラとインテルコンパイラ ver10.0＆Ver11.1 が高い。並列計算した場合ではコンパイラによる性能差はあまりない。マルチコアで計算したからといってコア数分計算が速くなる訳ではない。ベンチマークのサイズが大きい場合には、MPI、OpenMP、コンパイラの自動並列化機能、いずれの並列化を適用した場合にも性能はあまり変わらない。そういう意味では コンパイラの自動並列化機能は健闘していると言える。

4.2 メモリバンド幅とキャッシュ

シングルコアの CPU では、キャッシュを活用した計算が出来るかどうかが性能を左右していた。その状況はマルチコア CPU でも変わらない。マルチコア化によって CPU 全体としての演算性能が上がったため、メモリバンド幅に対する要求が大きくなっていて、結果としてキャッシュを活用できるかどうかが性能に大きく影響を与えている。特にここで測定した姫野ベンチマークのように、浮動小数点演算数に対する、メモリへのアクセス数の比率が高い場合には、その影響が大きい。ただし、マルチコア化によって、結果的に1CPU あたりで利用できるキャッシュの量が増えたため、シングルコアではキャッシュに載らなかった計算がマルチコアではキャッシュにおさまるようになり、性能が上がるということもある。

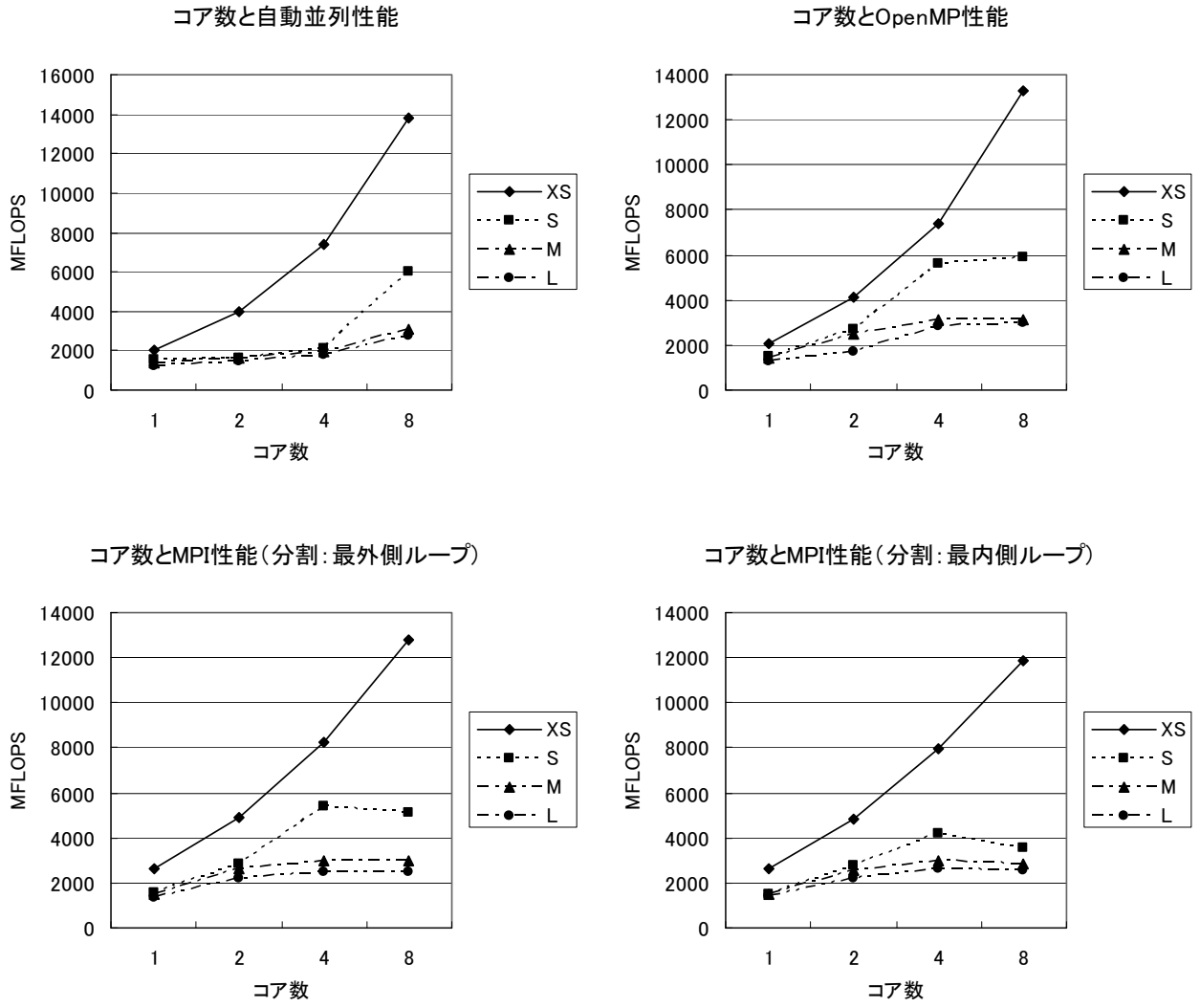

図 5. コア数と並列性能

以上

【補足資料】

■QX6700 のベンチマーク結果：サイズと MFLOPS 値（スカラー以外は 4 コア）

スカラー

サイズ	frt	ifort	gfortran
XS	1825.4	2076.06	914.64
S	1580.27	1536.02	859.49
M	1440.79	1509.78	787.01
L	1369.46	1429.59	710.71

自動並列

サイズ	frt	ifort
XS	6916.59	5903.31
S	2127.82	1914.47
M	1919.27	1819.46
L	1860.56	1837.81

OpenMP

サイズ	frt	ifort	gfortran
XS	7021.03	8213.53	3085.21
S	2141.4	2050.36	1975.97
M	1914.78	1874.47	1834.03
L	1859.85	1831.23	1790.23

MPI

サイズ	frt	ifort	gfortran
XS	5083.88	5332.59	4612.03
S	1803.73	1799.86	1792.8
M	1826.73	1827.02	1848.51
L	1825.07	1835.12	1854.96

■E5462 x2 でのベンチマーク結果：サイズと MFLOPS 値（スカラー以外は 8 コア）

スカラー

サイズ	frt	ifort	gfortran
XS	1993.35	2305.17	985.71
S	1512.17	1500.79	899.22
M	1415.78	1442.64	787.19
L	1247.71	1333.44	694.25

自動並列

サイズ	frt	ifort
XS	13653.96	14231.42
S	6570.77	5289.95
M	3550.24	3182.46
L	3140.34	2698.68

OpenMP

サイズ	frt	ifort	gfortran
XS	13285.28	15269.24	5462.88
S	5889.66	5795.5	4790.59
M	3159.33	3325.99	3020.31
L	2999.57	3068.13	2752.87

MPI

サイズ	frt	ifort	gfortran
XS	11897.21	11735.14	11393.37
S	3538.89	3568.11	3601.4
M	2826.37	2855.61	2851.65
L	2588.95	2625.89	2738.42

■Core i7 の 940 ベンチマーク結果：サイズと MFLOPS 値（スカラー以外は 4 コア）

スカラー

サイズ	frt	ifort	gfortran
XS	3784.19	4564.07	1295.31
S	3393.25	3982.34	1277.67
M	3220.94	3859.39	1181.88
L	2779.55	3431.87	1018.58

自動並列

サイズ	frt	ifort
XS	13279.37	13936.09
S	6295.6	5630.32
M	5968.83	5254.35
L	5745.72	5354.06

OpenMP

サイズ	frt	ifort	gfortran
XS	13571.87	17146.44	4535.93
S	6313.61	6013.17	4177.21
M	5935.83	5465.88	3849.17
L	5738.53	5498.23	3480.91

MPI

サイズ	frt	ifort	gfortran
XS	13252.81	11939.04	8861.22
S	5805.04	5801.21	5811.25
M	6126.84	6131.4	6158.61
L	6196.14	6193.41	6220.98

■E5462 x 2 でのベンチマーク結果：コア数と MFLOPS 値

自動並列

コア数	parallel (XS)	parallel (S)	parallel (M)	parallel (L)
1	2001.83	1513.36	1414	1217.98
2	3954.33	1617.49	1611.3	1488.37
4	7363	2112.21	1934.28	1769.42
8	13837.51	6036.84	3121.32	2758.49

OpenMP

コア数	OpenMP (XS)	OpenMP (S)	OpenMP (M)	OpenMP (L)
1	2073.09	1510.62	1414.12	1252.38
2	4089.02	2705.88	2517.85	1721.09
4	7376.55	5604.05	3159.14	2837.45
8	13285.28	5889.66	3159.33	2999.57

MPI
（分割は最外側ループ）

コア数	MPI (XS)	MPI (S)	MPI (M)	MPI (L)
1	2662.82	1542.29	1512.72	1358.4
2	4906.28	2843.86	2627.57	2207.25
4	8220.93	5394.65	2964.1	2497.33
8	12798.31	5134.03	2993.13	2491.56

MPI
（分割は最内側ループ）

コア数	MPI (XS)	MPI (S)	MPI (M)	MPI (L)
1	2661.75	1518.47	1514.34	1389.72
2	4861.83	2792.92	2584.6	2169.11
4	7934.51	4212.4	2950.75	2622.96
8	11895.4	3538.89	2817.83	2588.95