# 各種計算機基本性能調査

## 平成24年度第1四半期

目次


1. SR16000/M1 システム
   - 1.1 姫野ベンチ
   - 1.2 重力多体問題（N体問題）
   - 1.3 分子動力学計算
   - 1.4 ループ積分
2. BG/Q システム
   - 2.1 姫野ベンチ
   - 2.2 重力多体問題（N体問題）
   - 2.3 ループ積分

性能評価用プログラムとして特徴のある3つのものを使用しています。これらは並列化が容易という共通点があります。

（1）姫野ベンチプログラム
演算は単精度,演算量はメモリ量に比例。
3次元領域サイズを2049*2049*1025とすると、メモリ容量は192GB,1点の計算を行う際、16GB離れた領域を10回参照します。SIMD命令は適用されます。

（2）重力多体問題（N体問題）
演算は倍精度、演算量はメモリ量の2乗に比例。
粒子数をNとすると、メモリ容量は104N　バイト。
一般的なサイズはN=2^15=32768から2^20=1048576で
メモリサイズは3.328MBから104MBになります。
SIMD命令は適用されます。

（3）ループ積分
演算は4倍精度、演算量はメモリ量の積分次元数乗に比例。
二重指数関数型積分法で3次元積分の場合はテーブルサイズは,N=1024,2048,4096 でメモリ容量は32KB,64KB,128KBと
非常に小さくて済みます。SIMD命令は適用されません。

これらのジョブから並列実行時には、
領域分割が必要か否か
フラットMPIかMPIとSMPの組み合わせか
メモリ配置の変更が必要か否か
SMTの数は幾つが良いか（1，2，4）
を決める必要があります。

1．SR16000 モデルM1

SR16000/M1システムの複数ノードでの実行に関しての記述です。

1ノードの概略は以下の様になっています。

プロセッサ:power7
周波数:3.83GHz
CPUコア数 32(物理的),64(論理的)
理論最大性能　980.48 GFLOPs
メモリ容量 256GB
メモリアーキテクチャー　NUMA,(16論理コア単位でflat)
SIMD(Single Instruction Multiple Data)を
サポートするVSX機構付き
L3キャッシュ　On-Chip 32MB/8コア
演算器/物理コア　乗加算器4つ

## 1.1 姫野ベンチ

姫野ベンチは演算精度は単精度で核の部分は以下の様になっていて、SIMD命令は適用されます。

```
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)
   1              +a(I,J,K,3)*p(I,J,K+1)
   2              +b(I,J,K,1)*(p(I+1,J+1,K)-p(I+1,J-1,K)
   3              -p(I-1,J+1,K)+p(I-1,J-1,K))
   4              +b(I,J,K,2)*(p(I,J+1,K+1)-p(I,J-1,K+1)
   5              -p(I,J+1,K-1)+p(I,J-1,K-1))
   6              +b(I,J,K,3)*(p(I+1,J,K+1)-p(I-1,J,K+1)
   7              -p(I+1,J,K-1)+p(I-1,J,K-1))
   8              +c(I,J,K,1)*p(I-1,J,K)+c(I,J,K,2)*p(I,J-1,K)
   9              +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)
            WGOSA=WGOSA+SS*SS
            wrk2(I,J,K)=p(I,J,K)+OMEGA *SS
          enddo
        enddo
      enddo
```

ここで配列a,b,cのメモリアクセスで非常に離れた領域を参照する事になります。

もとの配列は以下の様に定められています。

```
CC Array
      dimension  p(mimax,mjmax,mkmax)
      dimension  a(mimax,mjmax,mkmax,4),
     >          b(mimax,mjmax,mkmax,3),c(mimax,mjmax,mkmax,3)
      dimension  bnd(mimax,mjmax,mkmax)
      dimension  wrk1(mimax,mjmax,mkmax),
      >wrk2(mimax,mjmax,mkmax)
```

これを,
A,b,c をまとめて配列aとして、以下の様にメモリ配置を変更すると、連続領域を参照する様になります。この場合にはSIMD命例は適用されなくなります。

```
CC Array
      dimension  p(mimax,mjmax,mkmax)
      dimension  a(10,mimax,mjmax,mkmax)
      dimension  bnd(mimax,mjmax,mkmax)
      dimension  wrk1(mimax,mjmax,mkmax),
     >wrk2(mimax,mjmax,mkmax)
```

ソースプログラムの名称は最初の場合をori,後者の場合をTune と表しています。

計算領域サイズを2049＊2049＊1025とすると、所要メモリ容量は192GBとなるので、複数ノード、領域分割をして実行しました。並列化はフラットMPIを使用しています。

| ソース | MPI数 | ノード数 | GFLOPs |
|---|---|---|---|
| ori | 512 | 8 | 269 |
| tune | 512 | 8 | 967 |
| ori | 256 | 8 | 166 |
| tune | 256 | 8 | 850 |
| ori | 256 | 4 | 135 |
| tune | 256 | 4 | 439 |
| ori | 128 | 4 | 87 |
| tune | 128 | 4 | 439 |

すべてのケースで配列の再配置の効果がでています。

8ノードでフラットMPIとSMP＊MPIのハイブリッドの比較は以下の様になりました。

| ソース | MPI数 | SMP数 | GFLOPs |
|---|---|---|---|
| ori | 512 | 1 | 269 |
| tune | 512 | 1 | 967 |
| ori | 256 | 1 | 166 |
| tune | 256 | 1 | 850 |
| ori | 8 | 64 | 227 |
| tune | 8 | 64 | 560 |
| ori | 8 | 32 | 146 |
| tune | 8 | 32 | 556 |

フラットMPIがハイブリッドより効果があります。

## 1.2 重力多体問題（N体問題）

重力多体問題は

$$\vec{F}_i = Gm_i \sum_{j \neq i} \frac{m_j}{r_{ij}^3} \vec{r}_{ij} \qquad r_{ij} = |\vec{r}_i - \vec{r}_j|$$

$G$：万有引力定数, $m_i$：粒子$i$の質量

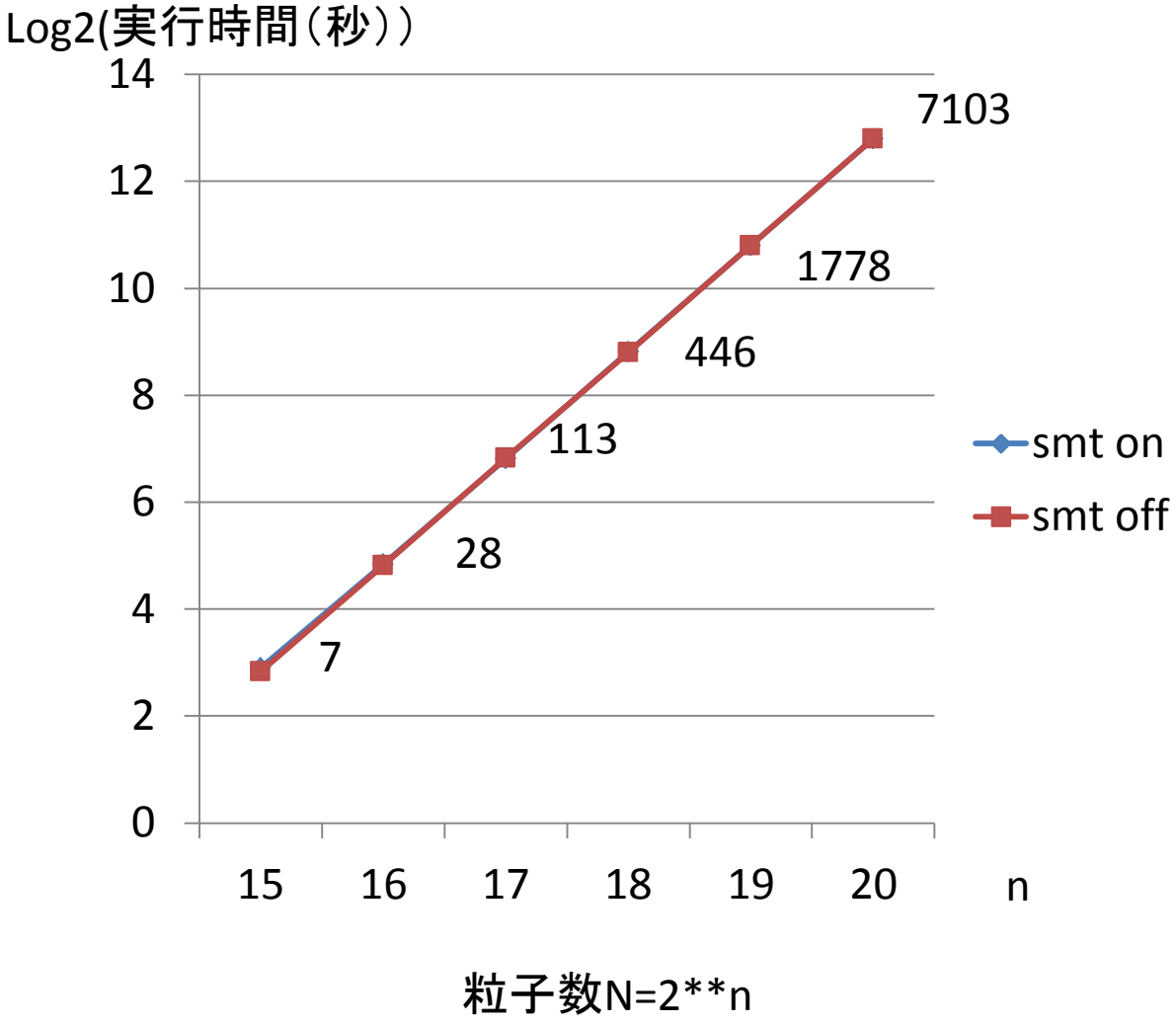
$\vec{r}_i$：粒子$i$の位置, $\vec{F}_i$：粒子$i$にかかる力

で計算します。

演算精度は倍精度で*SIMD*命令が適用されます。

演算量は粒子数Nの二乗に比例します。

4ノードでフラットMPIでSMT　ON、SMT　OFFの場合の結果です。
演算量がN=2**nの二乗に比例しますので、実行時間はlog2
スケールで表しています。タイムステップ数は100です。




SMT ON,OFF での実行時間に有意差は見られません。

また、フラットMPIとハイブリッドの差をN=2**18,2**20で調べた結果は以下の様になりました。

| サイズ2**n | smp数 | task数 | ノード数 | 実行時間(秒) |
|---|---|---|---|---|
| 20 | 1 | 256 | 8 | 3599 |
| 20 | 1 | 512 | 8 | 3612 |
| 20 | 32 | 8 | 8 | 3603 |
| 20 | 64 | 8 | 8 | 3604 |
| 20 | 8 | 32 | 8 | 3593 |
| 20 | 8 | 64 | 8 | 3607 |

| サイズ2**n | smp数 | task数 | ノード数 | 実行時間(秒) |
|---|---|---|---|---|
| 18 | 1 | 256 | 8 | 224 |
| 18 | 1 | 512 | 8 | 231 |
| 18 | 32 | 8 | 8 | 222 |
| 18 | 64 | 8 | 8 | 228 |
| 18 | 8 | 32 | 8 | 222 |
| 18 | 8 | 64 | 8 | 223 |

**フラットMPIとハイブリッドでも実行時間に有意差は見られません。**

## 1.3 分子動力学計算

分子動力学計算は以下の２つの計算からなり、
クーロン力計算は重力多体問題と同様の計算と
なりますが、異なる計算として、
ワンデルワールス力計算があります。逆に二乗則と
逆六乗則の差により、演算する範囲が異なり、演算量
も単純に粒子数の何乗に比例するとは言えません。
演算精度は倍精度でSIMD命令は一部適用されます。

クーロン力　$\vec{F}_i = \frac{q_i}{4\pi\varepsilon_0} \sum_{j \neq i} \frac{q_j}{r_{ij}^3} \vec{r}_{ij}$　　$r_{ij} = |\vec{r}_i - \vec{r}_j|$

$q_i$：電荷量, $\varepsilon_0$：真空の誘電率

ワンデルワールス力

分子間に働く分散力で、等方向性で原子間
距離の６乗に反比例する力

$F = k \times \frac{\alpha_a \alpha_b}{r^6}$　　$\alpha$：分極率

SR16000/M1　複数ノードでの実行時間は以下の様になっています。実行はフラットMPIを使用しています。所要メモリはそれぞれ、11.8MB,28.0MB,54.7MBとなっています。

| | 実行時間（単位秒） | | | |
|---|---|---|---|---|
| | | N=48 | | |
| MPI数 | ノード数 | VdW | Coulomb | 全体 |
| 128 | 4 | 1.8198 | 4.44435 | 6.26732 |
| 256 | 4 | 2.52998 | 8.61865 | 11.15244 |
| 256 | 8 | 1.71889 | 2.66658 | 4.38859 |
| 512 | 8 | 3.35877 | 3.98576 | 7.34833 |
| | | N=64 | | |
| MPI数 | ノード数 | VdW | Coulomb | 全体 |
| 128 | 4 | 12.54277 | 63.21624 | 75.76465 |
| 256 | 4 | 16.55753 | 97.01006 | 113.57417 |
| 256 | 8 | 8.77493 | 35.47022 | 44.25077 |
| 512 | 8 | 20.90441 | 63.69971 | 84.61077 |
| | | N=80 | | |
| MPI数 | ノード数 | VdW | Coulomb | 全体 |
| 128 | 4 | 44.93113 | 412.63567 | 457.57689 |
| 256 | 4 | 63.25216 | 362.52039 | 425.78437 |
| 256 | 8 | 33.06468 | 259.85763 | 292.93251 |
| 512 | 8 | 91.30541 | 281.84471 | 373.16153 |

粒子の数はn=N**3 で演算量はCoulmb(クーロン力)はn(n-1)/2に比例,VdW(ファンデルワールス力)はn(n-1)/2/近傍にある粒子の数に比例します。

VdWではすべてのケースでSMT OFF の効果があり、クーロン力はN=80,4node ではSMT　ONがそれ以外はSMT　OFFが効果があります。これは転送量がMPI数に比例するため、演算量との比でSMT　ON,OFFの効果が異なる様になります。

## 1.4 ループ積分

**測定したループ積分の積分式,解析近似解は以下の様なものです。**
**演算精度は4倍精度でSIMD命令は適用されません。**

$$I=\int_0^1\int_0^{1-x}\int_0^{1-x-y}\frac{1}{D^2}dzdydx$$

式

$$D=-sxy-tz(1-x-y-z)+(x+y)\lambda^2+(1-x-y-z)(1-x-y)m_e{}^2$$
$$+z(1-x-y)m_f{}^2$$

$$s=-500^2, t=-150^2, m_f=150,$$

データ

$$m_e=0.0005, \lambda=10^{-30}$$

**解析近似解**

$$I=\frac{1}{-s(-t+m_f{}^2)}\ln(\frac{-s}{\lambda^2})\ln\frac{(-t+m_f{}^2)^2}{m_e{}^2m_f{}^2}$$

相対誤差$7\times10^{-26}$以下

1ノードでサイズN=1024,2048,4096での実行結果です。
ここでは、SMT数＝SMP数/32と記しています。

N=1024

| smp数 | smt数 | 実行時間（秒） |
|---|---|---|
| 32 | 1 | 4.036747 |
| 64 | 2 | 2.396178 |
| 96 | 3 | 3.429908 |
| 128 | 4 | 3.842133 |

N=2048

| smp数 | smt数 | 実行時間（秒） |
|---|---|---|
| 32 | 1 | 32.350153 |
| 64 | 2 | 19.282983 |
| 96 | 3 | 26.350305 |
| 128 | 4 | 27.733024 |

N=4096

| smp数 | smt数 | 実行時間（秒） |
|---|---|---|
| 32 | 1 | 256.204864 |
| 64 | 2 | 153.281336 |
| 96 | 3 | 205.601229 |
| 128 | 4 | 195.04127 |

演算が4倍精度のため、SIMD命令が適用されないので、SMP数＝64、SMT数＝2（SMT　ON)の場合がもっとも高速になります。

複数ノード,フラットMPIでのN=1024,2048,4096の場合の実行時間は以下の様になりました。

N=1024

| MPI数 | ノード数 | 実行時間（秒） |
|---|---|---|
| 128 | 4 | 1.009251 |
| 256 | 4 | 0.60237 |
| 256 | 8 | 0.50608 |
| 512 | 8 | 0.302202 |

N=2048

| MPI数 | ノード数 | 実行時間（秒） |
|---|---|---|
| 128 | 4 | 8.069988 |
| 256 | 4 | 4.787942 |
| 256 | 8 | 4.047229 |
| 512 | 8 | 2.394252 |

N=4096

| MPI数 | ノード数 | 実行時間（秒） |
|---|---|---|
| 128 | 4 | 64.550396 |
| 256 | 4 | 38.031384 |
| 256 | 8 | 32.274641 |
| 512 | 8 | 19.017973 |

演算量がNの三乗に比例する事と、SMT　ONの効果がでる事がはっきりとでています。

## ２　BG/Q　システム

使用していますBG/Q　システムの概略は以下の様になっています。

周波数　1.6GHz
1 ノード　16core　論理性能　204.8GFLOPs
L1 キャシュ　　16/16KB (Core)
L2　32MB (node)
Main storage　16GB　(Core)
Smt=1,2,4

実行クラス
　　32 node　6553.6 GFLOPs
　　128 node　26214.4 GFLOPs
　　256 node　52428.8 GFLOPs
　　512　node　1048576.6 GFLOPs
あり、今回は32 node を中心にした測定結果をしめします。

## ２．１　姫野ベンチ

領域サイズ2049*2049*1025 で所要メモリは１９２GB、
１要素の演算中１６GB離れた領域を１０回アクセスします。

SMT数と配列のメモリ再配置の効果の関係の調査結果は
以下の様になりました。

| 3次元分割 | ソース | MPI数 | ノード数 | SMT数 | GFLOPs |
|---|---|---|---|---|---|
| 32*16*16 | ori | 8192 | 128 | 4 | 975 |
| 32*16*16 | tune | 8192 | 128 | 4 | 1,065 |
| 16*16*16 | ori | 4096 | 128 | 2 | 695 |
| 16*16*16 | tune | 4096 | 128 | 2 | 1,135 |
| 16*16*8 | ori | 2048 | 128 | 1 | 842 |
| 16*16*8 | tune | 2048 | 128 | 1 | 811 |

１ノード当たりの所要メモリは１．５GB、１スレッドあたり
128MPI　で2MB,4096MPI で4MB,2048MPIで8MB離れた領域
をアクセスします。配列のメモリ再配置の影響はSMT=2が最も
大きく,SMT=1ではもとの配列のメモリ配置のほうが良い数値
となっています。これは、もとの配列のメモリ配置ではSIMD命令
が適用され、配列のメモリ再配置だとSIMD命令が適用されない
事とL2キャシュ2MBである影響によるものです。
総合的にみると、SMT数が4の場合がもっとも高速といえます。

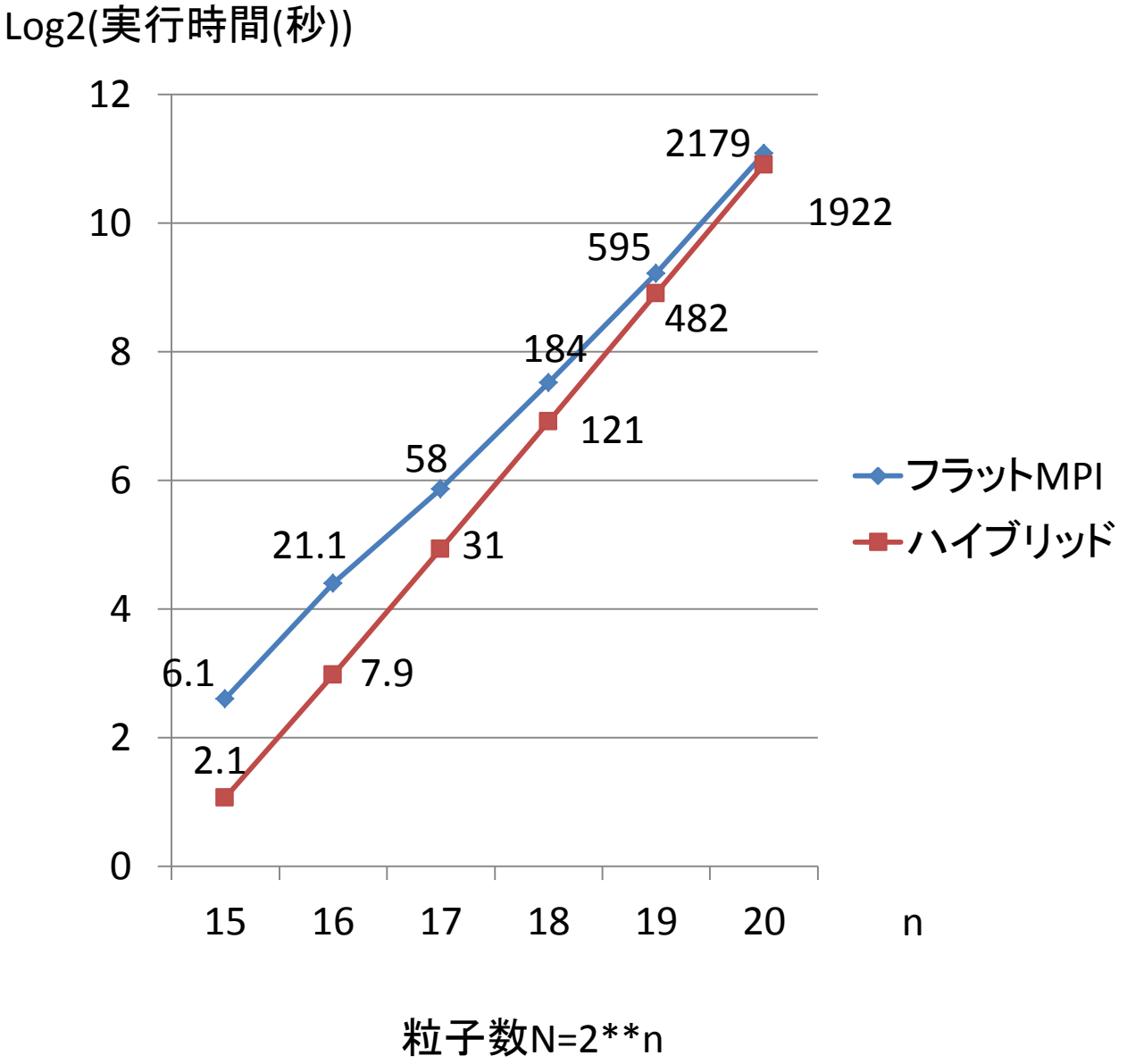
さらにノード数を増やし、SMT=4の場合の結果は以下の様になりました。

| 3次元分割 | ソース | MPI数 | ノード数 | SMT数 | GFLOPs |
|---|---|---|---|---|---|
| 32*32*32 | ori | 32768 | 512 | 4 | 4,163 |
| 32*32*32 | tune | 32768 | 512 | 4 | 5,038 |
| 16*32*32 | ori | 16384 | 512 | 2 | 2,139 |
| 16*32*32 | tune | 16384 | 512 | 2 | 4,462 |

所要メモリは1ノード当たり384MB,MPI数32768で0.5MB,MPI数16384で1MB離れた領域をアクセスするため、L2　2MBより配列のメモリ再配置の効果の様子が大きく変わっています。

## ２．２　重力多体問題（N体問題）

領域分割をせずに並列実行する場合、粒子数N=2**15=32768からN=2**20=1048576 とすると1スレッド当たりの所要メモリ量が3.328MBから104MBため、SMPとMPIのハイブリッド方式がフラットMPIより効果がでます。演算量がNの二乗に比例するため、実行時間の軸はlog2を取っています。実行結果は以下の様になりました。タイムステップ数は１００です。



## ２．３　ループ積分

1ノードでサイズN=1024,2048,4096での実行結果です。
ここでは、SMT数＝SMP数/16となります。

N=1024

| smp数 | smt数 | 実行時間（秒） |
|---|---|---|
| 16 | 1 | 31.759181 |
| 32 | 2 | 17.368297 |
| 48 | 3 | 12.878649 |
| 64 | 4 | 11.529416 |

N=2048

| smp数 | smt数 | 実行時間（秒） |
|---|---|---|
| 16 | 1 | 254.202281 |
| 32 | 2 | 138.318369 |
| 48 | 3 | 101.279492 |
| 64 | 4 | 91.819282 |

N=4096

| smp数 | smt数 | 実行時間（秒） |
|---|---|---|
| 16 | 1 | 2032.887439 |
| 32 | 2 | 1106.144559 |
| 48 | 3 | 808.601665 |
| 64 | 4 | 734.234379 |

演算が4倍精度のため、SIMD命令が適用されないので、
SMP数＝64、SMT数＝4の場合がもっとも高速になります。

複数ノード,フラットMPIでのN=1024,2048,4096の場合の実行時間は以下の様になりました。

N=1024

| MPI数 | ノード数 | 実行時間（秒） |
|---|---|---|
| 1024 | 32 | 0.524877 |

N=2048

| MPI数 | ノード数 | 実行時間（秒） |
|---|---|---|
| 512 | 32 | 7.923729 |
| 1024 | 32 | 4.212413 |
| 2048 | 32 | 2.838767 |

N=4096

| MPI数 | ノード数 | 実行時間（秒） |
|---|---|---|
| 512 | 32 | 84.754495 |
| 1024 | 32 | 44.105198 |
| 2048 | 32 | 28.051443 |

演算量がNの三乗に比例していますが、所要メモリは1ノード当たり、N=1024で1MB,N=2048で2MB,N=4096で4MBとなり、N=2048とN=4096では演算量比以上の実行時間比となっています。