各種計算機基本性能調査

平成26度第1四半期

目次

1. はじめに

数値計算において以下のテーマを定めました。

1.GPUの適用
2.データ形式の差の影響
3.FORTRANとCの比較
4.多次元数値積分
5.量子モンテカルロ法による物性スペクトル
　計算

これらを、計算機の特徴に応じてプログラムを
実行しました。

SR16000/M1 システムでは、性能評価の基準
となる演算量の算出と倍精度,4倍精度の
多次元積分を中心に、サーバーでは整数演算方式
による超多倍長計算、FORTRANとCの精度比較,
GPUではFORTRANとCの性能評価を
行っています。

2.性能評価問題の演算量

今回は主にGPUで演算するケースを扱っている
ためその演算量をSR16000の性能モニター
より算出しています。ただしGPU等での
拡張倍精度演算や4倍精度演算は省いて
います。
4倍精度演算では倍精度+倍精度の演算量
を参考までに算出しています。
SR16000でieee754-2008形式4倍精度
では非常に実行時間がかかる事と性能モニター
では演算量は0 (MFLOPs) となるため,

CPU time	Flop	Inst	MFLOPS	MIPS	Times
225.901	0	46460M	0.000	6211.925	32

今回は扱いませんでした。
将来的には整数演算を用いたシュミレーションも
ありますので (クラスターのシュミレーション等)
評価方法 (MIPS等)を検討予定です。

2.1 行列積計算

行列積は定義式どうりのコーディングで
(アンローリングやキャシュチューニングなし) で,
1ノード64smpの場合の値で算出しています。
4倍精度は自動コンパイルで,6,8,10倍精度は
人手でuinline化したものを計測しています。

行列積	演算量(GFLOP)	
	1000回実行	
精度	N=240	N=480
4倍精度	470	3760
6倍精度	1849	14789
8倍精度	3866	30928
10倍精度	8657	69265

2.2 QDR積計算

Q(行列), D(ベクトル), R(行列)

$$C_{i,j} = \sum_{k=1}^{n} Q_{i,k} (DR)_{k,j}$$

$$(DR)_{k,j} = D_k R_{k,j} (j = 1,...., n)$$

量子モンテカルロ法による物性スペクトル計算で4倍精度演算で精度の良い結果が得られなかった事もあり, 演算量は実際のケースに即し,6,8,10倍精度演算の場合を算出しています。

QDR積計算	1000回実行の演算量(GFLOP)	
精度	n=240	n=480
6倍精度	2878	23026
8倍精度	5969	47755
10倍精度	13102	104823

2.3 多次元積分計算

(1)4次元積分計算

$$I=\int_0^1\int_0^{1-x}\int_0^{1-x-y}\int_0^{1-x-y-z}\frac{1}{DC}dudzdydx$$

$$C=(x+y+z+u)(1-x-y-z-u)+(x+y)(z+u)$$

$$E=(1-x-y-z-u)(x+z)(y+u)+(x+y)zu+(z+u)xy$$

$$M^2=xm_1^{\ 2}+ym_2^{\ 2}+zm_3^{\ 2}+um_4^{\ 2}+(1-x-y-z-u)m_5^{\ 2}$$

$$D=-sE+M^2C$$

で変数変換により,積分区間を$[0,1]^4$にして4重DOループのものをcase1,ループ併合して2重DOループにしたものをcase2としてサイズは$N=576$にしています。

(2)5次元積分

$$I=\int_0^1\int_0^{1-x_1}\int_0^{1-x_1-x_2}\int_0^{1-x_1-x_2-x_3}\int_0^{1-x_1-x_2-x_3-x_4}\frac{1}{D^2}dx_5dx_4dx_3dx_2dx_1$$

$$x_6=1-x_1-x_2-x_3-x_4-x_5$$

を変数変換で積分区間$[0,1]^5$にして,5重DOループを外側DOループ(2つのDOループを併合)×内側DOループ(3つのDOループを併合)で,$N=120$, Dの内容でcase1, case2としています。

5次元case1

```
D =-x1**2*x2-x1**2*x3-x1**2*x4-x1**2*x6-x1*x2**2-x1*x2*x3
   &-2.d0*x1*x2*x4
   &-x1*x2*x5-x1*x2*x6-x1*x3**2-2.d0*x1*x3*x4-x1*x3*x5-x1*x3*x6
   &-x1*x4**2
   &-x1*x4*x5-2.d0*x1*x4*x6-x1*x5*x6-x1*x6**2- x2**2*x4-x2**2*x5
   &-x2*x3*x4
   &-x2*x3*x5-x2*x4**2-2.d0*x2*x4*x5-x2*x4*x6-x2*x5**2-x2*x5*x6
   &-x3**2 *x4
   &-x3**2*x5-x3*x4**2-2.d0*x3*x4*x5-x3*x4*x6-x3*x5**2-x3*x5*x6
   &-x4**2*x5
   &-x4**2*x6-x4*x5**2-3.d0*x4*x5*x6-x4*x6**2-x5**2*x6-x5*x6**2
```

5次元case2

```
D=-x1*x1*x2-x1*x1*x4-x1*x1*x5-x1*x1*x6-x1*x2*x2-x1*x2*x3
   .-x1*x2*x4
   .-2.0d0*x1*x2*x5-2.0d0*x1*x2*x6-x1*x3*x4-x1*x3*x5-x1*x3*x6
   .-x1*x4*x4-3.0d0*x1*x4*x5
   .-2.0d0*x1*x4*x6-x1*x5*x5-x1*x5*x6- x1*x6*x6-x2*x2*x3
   .-x2*x2*x5-x2*x2*x6
   .-x2*x3*x3-x2*x3*x4-x2*x3*x5-2.0d0*x2*x3*x6-x2*x4*x5
   .-x2*x4*x6-x2*x5*x5-x2*x5*x6-x2*x6*x6-x3*x3*x4-x3*x3*x5
   .-x3*x3*x6-x3*x4*x4-2.0d0*x3*x4*x5-2.0d0*x3*x4*x6-x3*x5*x5
   .-x3*x5*x6-x3*x6*x6-x4*x4*x5-x4*x4*x6-x4*x5*x5-x4*x5*x6
   . -x4*x6*x6
```

(3)6次元積分

変数変換により積分区間を$[0,1]^6$にして3重DOループをひとまとめにした2重DOループを作成。サイズはN＝120.

以下の3つの問題を選択しています。

6次元case1

$$I=\int_0^1\int_0^{1-x_1}\int_0^{1-x_1-x_2}\int_0^{1-x_1-x_2-x_3}\int_0^{1-x_1-x_2-x_3-x_5}\int_0^{1-x_1-x_2-x_3-x_5-x_6}\frac{C}{D^3}dx_7dx_6dx_5dx_3dx_2dx_1$$

$$x_4=1-x_1-x_2-x_3-x_5-x_6-x_7$$

```
C=x1*x4+x1*x5+x1*x6+x2*x4+x2*x5+x2*x6+x3*x4+x3*x5+x3*x6+x4*x5
    &   + x4*x6+x4*x7+x5*x7+x6*x7

    D =-(x1**2+x2**2+x3**2+x7**2+x1*x2+x1*x3+x1*x7+x2*x3+x2*x7+x3*x7)
    &*(x4+x5+x6)
    &-x4**2*(x1+x2+x3+x5+X6+x7)
    &-(x5**2+x6**2+x5*x6)*(x1+x2+x3+x4+x7)
    &-3.0*x4*(x1*x5+x6*x7)
    &-2.0*((x1+x2+x3)*x4*x6+(x2+x3+x7)*x4*x5)
```

6次元case2

$$I=\int_0^1\int_0^{1-x_1}\int_0^{1-x_1-x_3}\int_0^{1-x_1-x_3-x_2}\int_0^{1-x_1-x_3-x_2-x_7}\int_0^{1-x_1-x_3-x_2-x_7-x_6}\frac{C}{D^3}dx_4dx_6dx_7dx_2dx_3dx_1$$

$$x_5=1-x_1-x_3-x_2-x_7-x_6-x_4$$

```
C=(x1+x2+x3+x4)*(x4+x5+x6+x7)-x4*x4
    cc=x1*m12+x2*m22+x3*m32+x4*m42+x5*m52+x6*m62+x7*m72
    D=-c*cc
  .+s*(x1*x2*(x4+x5+x6+x7)+x5*x6*(x1+x2+x3+x4)+x1*x4*x6+x2*x4*x5)
  .+t*x3*x4*x7
  .+p12*(x1*x3*(x4+x5+x6+x7)+x3*x4*x5)
  .+p22*(x2*x3*(x4+x5+x6+x7)+x3*x4*x6)
  .+p32*(x5*x7*(x1+x2+x3+x4)+x1*x4*x7)
  .+p42*(x6*x7*(x1+x2+x3+x4)+x2*x4*x7)
```

6次元case3

$$I=\int_0^1\int_0^{1-x_1}\int_0^{1-x_1-x_2}\int_0^{1-x_1-x_2-x_3}\int_0^{1-x_1-x_2-x_3-x_4}\int_0^{1-x_1-x_2-x_3-x_4-x_5}\frac{C}{D^3}dx_6dx_5dx_4dx_3dx_2dx_1$$

$$x_7=1-x_1-x_2-x_3-x_4-x_5-x_6$$

```
c=(x1+x2+x3+x4+x5)*(x1+x2+x3+x6+x7)-(x1+x2+x3)*(x1+x2+x3);

cc=x1*m12+x2*m22+x3*m32+x4*m42+x5*m52+x6*m62+x7*m72;
   d=-c*cc
  +s*(x1*x2*(x4+x5+x6+x7)+x1*x5*x6+x2*x4*x7-x3*x4*x6)
  +t*x3*(-x4*x6+x5*x7)
  +p12*(x1*x3*(x4+x5+x6+x7)+x3*x4*(x6+x7))
  +p22*(x2*x3*(x4+x5+x6+x7)+x3*x6*(x4+x5))
  +p32*(x4*x5*(x1+x2+x3+x6+x7)+x4*x6*(x2+x3)+x1*x5*x7)
  +p42*(x6*x7*(x1+x2+x3+x4+x5)+x4*x6*(x1+x3)+x2*x5*x7);
```

多次元積分演算量(GFLOP)

次元	case	演算量
4次元	1	2976
4次元	2	4568
5次元	1	2514
5次元	2	2638
6次元	1	312064
6次元	2	243328
6次元	3	232896

4次元積分で多重Doループを一重化すると(case2) 演算量が5割増しとなっています。

3.倍精度多次元積分計算

3.1　4次元積分計算

	実行時間一覧表(秒)		
case	64smp	32core	演算量(GFLOP)
case1	11.645035	11.698903	2976
case2	24.126208	30.395169	4568

多重DOループを一重化すると演算量は
1.5倍に対し、実行時間は２～３倍とな
っています。これはSR16000のみの特徴です。

3.2 5次元積分計算

5次元積分実行時間一覧表(秒)			
case	32core	64smp	演算量(GFLOP)
case1	8.404	5.426	2514
case2	9.841	6.373	2638

5次元積分は積分変数割り当てが同じでDの式（演算量）で区別している事になるので、実行時間と演算量は良く比例しています。

32core（smt=off）と64smp（smt=on）の性能差が1.5倍あり、４次元積分とは少し違った傾向をしめしています。

3.3　6次元積分計算

	6次元積分実行時間一覧表(秒)		
case	32core	64smp	演算量(GFLOP)
case1	2223.744	1418.553	312064
case2	1929.049	1256.823	243328
case3	2212.909	1408.712	232896

積分変数の割り当てが異なるため、演算量と実行時間の関係があまりない。
Case1とcase3では演算量に30%程差があるのに実行時間はほぼ同じ。Case2とcase3は演算量と実行時間の大小が逆になっています。

Smt=offとsmt=onの実行時間の比が4,5次元積分に比べて大きくなっています。