各種計算機アプリケーション性能比較

平成26年度第1四半期

目次

1.はじめに

今までと少し性質の異なるグラフィックボードが使用できる様になったので従来のアプリケーションで性能比較を実施しました。

主に使用した計算機は以下のものです。

(1)SR16000/M1

プロセッサ:power7
周波数:3.83GHz
１ノード当たり
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

(2) GPGPU

GPU カード型番:ATI RadeonHD5870

メモリ: GDDR5, 1 GB, 153.6 GB/s

ホストインタフェース: PCI Express 2.1 x16stream processing unit: 3200個 (演算プロセッサ)

動作周波数: 850 MHzピーク性能 (単精度) :
5440 Gflops
(=3200x2x850MHz) ピーク性能 (倍精度) :
1088 Gflops

(3) グラフィックボード

HOST

E5－2670　2.60GHz　1cpu＝8core

キャシュ　20MB

2.6GHz×8×8＝166.4GFLOPs　2cpu

2cpu＝332.8GFLOPs

Xeon　Phi5110P　1.053GHz

60コア,4スレッド/1core

1.053GHz×60×4×4＝1010.88GFLOPs

2 行列積計算

比較はSR16000 1ノード(32core,64smp),ES-2670 16smp,
Phi5110P 240core　で実施しています。最も多くのケースを
実施したのは行列サイズN=240,N=480の1000回実行です。
Cは,ES-2670,Phi5110Pで倍精度,拡張倍精度,拡張倍精度
＋拡張倍精度のケースを行っています。名称はそれぞれ
Double,exdouble,ddexdoubleとしています。FORTRANは
SR16000,ES-2670,Phi5110Pで倍精度multd,4倍精度(ieee754
-2008データ形式)multq,dd形式4倍精度ddmultd,dd形式の
6倍精度,8倍精度,10倍精度,それぞれmult6,mult8,mult10
です.SR16000では特にmultd,multqは実施していません。
また比較という事で,テストプログラムは定義式どうりの
コーディングでアンローリング,キャシュチューニング等は
実施していません。演算量は以下の様になっています。

演算量(GFLOP)一覧

プログラム	N=240	N=480
multd	28	221
ddmultd	470	3760
mult6	1849	14789
mult8	3866	30928
mult10	8657	69265

行列サイズN=240の結果は以下の様になっています。

行列積N=240,1000回実行の性能比較				
	実行時間(秒)			
プログラム	SR16000		phi5110P	E5-2670
	32core	64smp	240smp	16smp
multd			1.824	0.403
ddmultd	7.360	3.753	2.952	5.618
multq			54.771	26.297
double			1.941	0.421
exdouble			5.735	2.663
ddexdouble			71.820	55.966
mult6	21.274	14.789	20.972	26.144
mult8	37.599	24.575	33.496	56.351
mult10	79.776	53.658	86.748	210.930

(1) dd形式4倍精度はphi5110Pは高速
(2) dd形式6倍精度,8倍精度,10倍精度はSR16000 32coreとphi5110Pの性能が同等。
(3) Cとソフトウエアサポートルーチンは E5-2670が高速。

行列サイズN=480の結果は以下の様になっています。

	行列積N=480,1000回実行の性能比較			
		実行時間(秒)		
プログラム	SR16000		phi5110P	E5-2670
	32core	64smp	240smp	16smp
multd			3.445	3.204
ddmultd	55.214	29.658	11.212	44.669
multq			469.772	209.251
double			3.728	3.196
exdouble			31.246	21.076
ddexdouble			256.872	191.585
mult6	164.514	119.144	159.443	205.080
mult8	289.634	196.611	285.541	460.643
mult10	605.880	436.391	688.837	1686.164

性能比較の傾向はN=240の場合と全く同じ。

各種行列積計算の性能は以下の様になっています。

各種行列積計算性能比較表

実行時間一覧表(秒)

N	CPU	double	exdouble	ddexdouble	multd	ddmultd
240	E5-2670	0.421	2.663	55.966	0.403	5.618
	Phi5110P	1.941	5.735	71.820	1.824	2.952
480	E5-2670	3.196	21.076	191.585	3.204	44.669
	Phi5110P	3.728	31.246	256.872	3.445	11.212
720	E5-2670	17.364	70.782	485.278	19.230	150.859
	Phi5110P	11.141	103.516	948.543	10.315	35.615
960	E5-2670	40.647	204.819	1035.046	45.322	357.364
	Phi5110P	25.932	253.483	2284.650	25.001	116.462
1200	E5-2670	79.576	369.371	1890.814	88.222	720.564
	Phi5110P	56.584	564.900	4642.316	57.053	241.207

(1) 倍精度ではc,FORTRANともN=720でPhi5110P　の性能がE5-2670を上回ります。
(2) 拡張倍精度はすべてE5-2670の性能が上回っています。
(3) SR16000の4倍精度では,Phi5110Pが非常に効力を発揮する事がわかります。

これに関連した計算は以下の様になっています。

4倍精度行列積計算

N=256　1回の実行時間

SR16000 singleジョブ
92.638 msec　ieeequad 8704.767 msec

x5570
double+double　457.259 msec
extend double +extend double 734.146 msec

e5430 gcc
double+double　652.392 msec
extend double +extend double 759.277 msec

e5430 icc
double+double　309.527 msec
extend double +extend double 621.636 msec

Phi5110P　240smpがSR16000 1ノードより高速という結果と合わせPhi5110Pが有効な事がわかります。

正しく計算するには仮数部121ビット以上必要で,演算順序も厳密に保証する必要があるrump’sの例題の結果です。

	rump's例題			
		100,000,000回実行		
			実行時間(秒)	
	精度	E5-2670	Phi5110P	
		16smp	240core	
	拡張倍精度*2	3.644167	26.985987	
	6倍精度	20.833355	249.228187	
	8倍精度	47.802169	582.083727	

E5-2670　拡張倍精度＋拡張倍精度が非常に高速。dd形式の4倍精度では指数部の制限の影響を受けるプログラムには,E5-2670で拡張倍精度＋拡張倍精度を使用するのが非常に有効です。

3.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，演算量の主要部分を占める6,8,10倍精度演算のQDR積計算で性能を比較しました。演算量は以下の様になっています。

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		

QDR積計算実行時間(秒)一覧表

N=240,1000回実行

CPU	SR16000		Phi5110P	E5-2670
精度	32core	64smp	240core	16smp
6倍精度	34.628	21.655	58.793	48.072
8倍精度	66.474	41.406	85.431	90.617
10倍精度	110.507	70.498	171.588	339.159

N=480,1000回実行

CPU	SR16000		Phi5110P	E5-2670
精度	32core	64smp	240core	16smp
6倍精度	259.771	172.937	475.308	382.709
8倍精度	499.358	331.659	686.868	745.481
10倍精度	842.099	573.847	1350.565	2713.501

SR16000　64smp (smt=on) の効果が大。Phi5110PとE5-2670では6倍精度と8倍精度での性能が逆転していますすま。これは再内側DOループ内の演算量が多くなるとPhi5110Pが効果を発揮する事によります。

4. N体問題計算

専用計算機,GPU等で非常に性能のでる
N体問題で性能を比較しました。

N 体問題
N ＝ 1000 ，タイムステップ　10000
演算量は約　500 GFLOP

実行時間一覧表(秒)

Phi5110P

core数	倍精度	拡張倍精度
25	7.723330	86.568028
50	5.507729	44.173921
100	3.854595	39.357228
200	2.888737	35.459120
240	2.788607	32.046642

E5-2670

smp数	倍精度	拡張倍精度
5	21.297323	34.762394
10	16.298939	20.509124
16	16.234552	16.206062

HD5880

gpu面	倍精度
一面 0	24.055599
一面 1	23.989969
二面	32.904664

Phi5110Pでは倍精度は非常に高速です。
倍精度と拡張倍精度の性能比はPhi5110P
は大きく,E5-2670は小さいという
差があります。HD5880は一面1600core,二面3200core
のため,粒子数1000ではあまり性能は出ていません。

HD5880で演算量が一定になる様に粒子数Nを変化させて測定しました。

N 体問題
粒子数 N , タイムステップ数 M
N^2M を一定になる様にしました。
演算量は約 500 GFLOP
HD 5880 の測定結果

実行時間一覧表(秒)

N=1000,タイムステップ10000

gpu面	倍精度
一面 0	24.055599
一面 1	23.989969
二面	32.904664

N=4000,タイムステップ 625

gpu面	倍精度
一面 0	4.612560
一面 1	4.601385
二面	5.240678

N=10000,タイムステップ 100

gpu面	倍精度
一面 0	2.747958
一面 1	2.746373
二面	1.991692

粒子数Nが10000以上で性能が出る様になっている事がわかります。

5.多次元積分計算

(1)2次元積分は

inf ra vtx

$$I = \int_0^1 \int_0^{1-x} \frac{1}{D} dydx$$

$$D = -xys + (x+y)^2 m^2 + (1-x-y)\lambda^2$$

$s = -500^2, m = 0.0005, \lambda = 10^{-5}$ で実施しています。

(2)3次元積分は

inf ra box

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

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

$s = -500^2, t = -150^2, m_f = 150, m_e = 0.0005$

$\lambda = 10^{-5}$ で実施しています。

演算量は従来との比較よりソース上からカウントしています。

(3)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。**

(4)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
```

(5)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

$$I = \int_0^1 \int_0^{1-x_1} \int_0^{1-x_1-x_3} \int_0^{1-x_1-x_3-x_2} \int_0^{1-x_1-x_3-x_2-x_7} \int_0^{1-x_1-x_3-x_2-x_7-x_6} \frac{C}{D^3} dx_4 dx_6 dx_7 dx_2 dx_3 dx_1$$

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

```
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

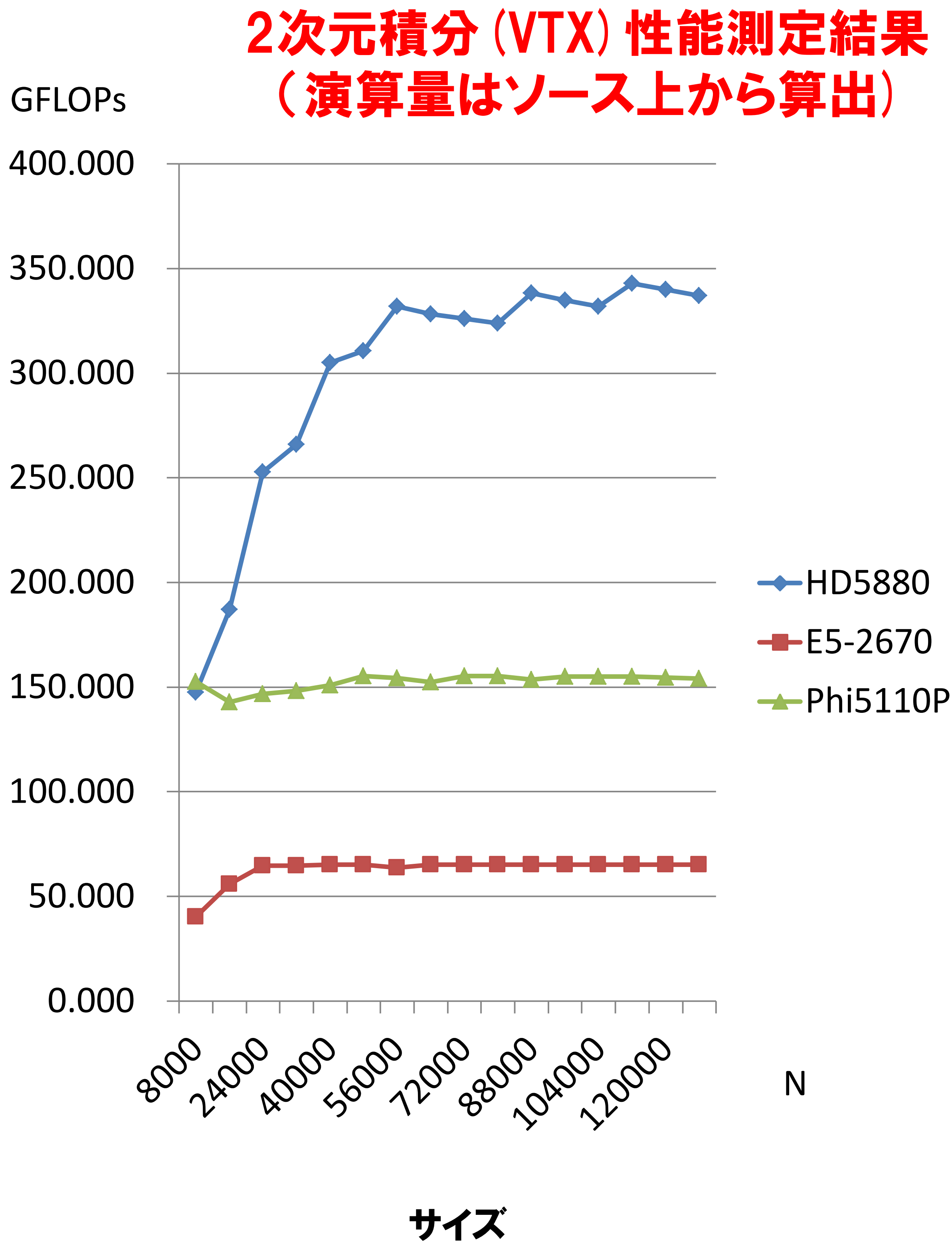
$$I = \int_0^1 \int_0^{1-x_1} \int_0^{1-x_1-x_2} \int_0^{1-x_1-x_2-x_3} \int_0^{1-x_1-x_2-x_3-x_4} \int_0^{1-x_1-x_2-x_3-x_4-x_5} \frac{C}{D^3} dx_6 dx_5 dx_4 dx_3 dx_2 dx_1$$

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

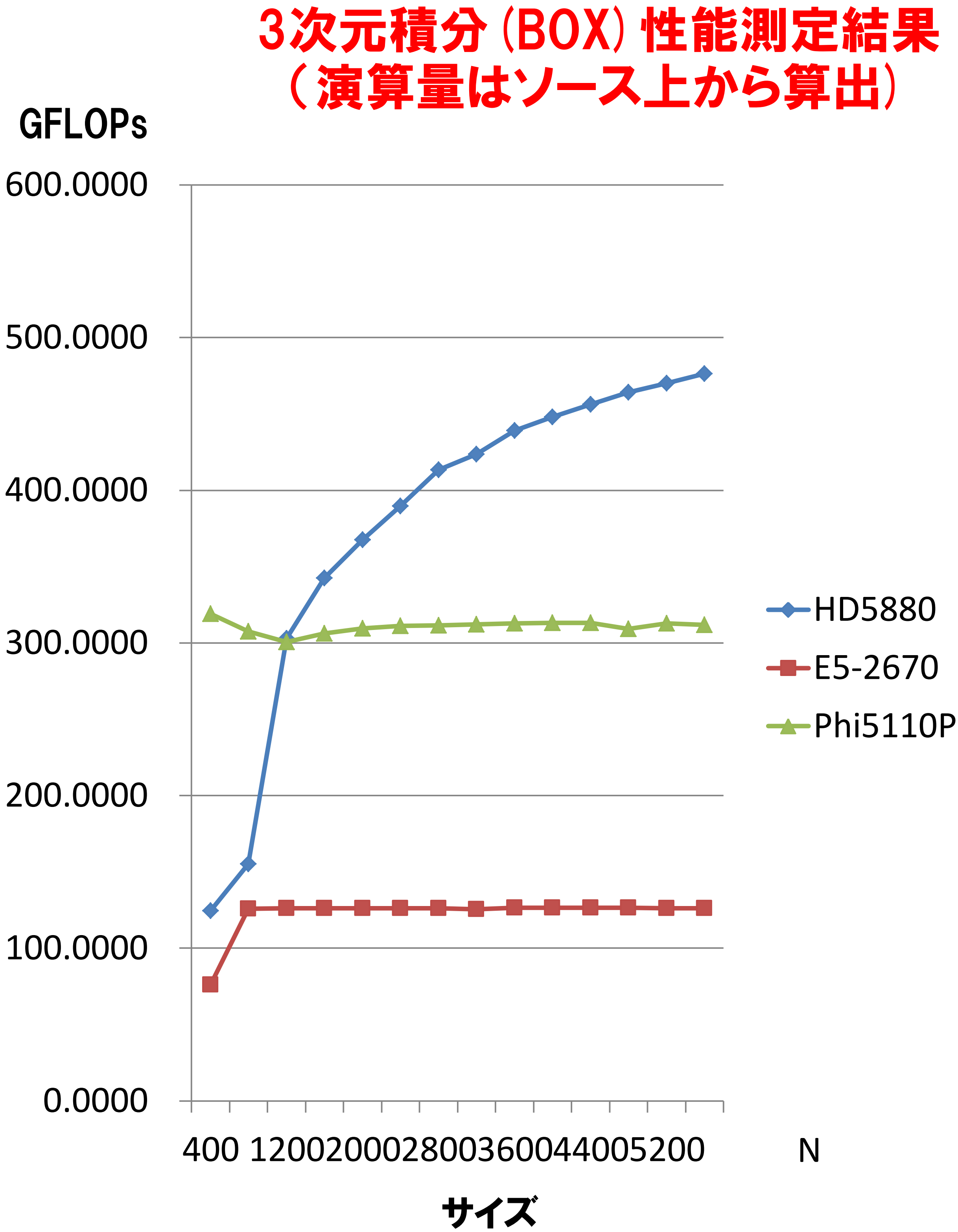
```
c=(x1+x2+x3+x4+x5)*(x1+x2+x3+x6+x7)-(x1+x2+x3)*(x1+x2+x3);

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);
```

5.1 2次元積分計算


2次元積分（VTX）性能測定結果
（演算量はソース上から算出）
GFLOPs
400.000
350.000
300.000
250.000
200.000
150.000
100.000
50.000
0.000
8000
24000
40000
56000
72000
88000
104000
120000
N
HD5880
E5-2670
Phi5110P
サイズ

5.2 3次元積分計算


3次元積分 (BOX) 性能測定結果
（演算量はソース上から算出）
GFLOPs
600.0000
500.0000
400.0000
300.0000
200.0000
100.0000
0.0000
400 1200 2000 2800 3600 4400 5200
N
サイズ
HD5880
E5-2670
Phi5110P

5.3 4次元積分計算

SR16000,E5-2670,Phi5110Pの比較の結果は以下の様になっています。

	4次元積分実行時間一覧表(秒)					
case	精度	SR16000		E5-2670	Phi5110P	演算量
		32core	64smp			((GFLOP)
case1	倍精度	11.698903	11.645035	27.618532	17.836520	2976
case1	拡張倍精度			108.657508	257.571165	
case2	倍精度	24.126208	30.395169	26.958704	17.815714	4568
case2	拡張倍精度			108.635537	256.691088	
case2	4倍精度			4538.965000	9762.636500	

(1) 多重DOループの一重化はE5-2670, Phi5110Pは若干速くなりますがSR16000は2~3倍程度遅くなっています。
(2) ieee754-2008形式の4倍精度は非常に遅くなっています。

サイズを大きくした場合とHD5880との比較は以下の様になっています。

4次元積分計算詳細テスト結果

case2	N=1200	実行時間(秒)
精度	E5-2670	Phi5110P
	16smp	240core
倍精度	506.127663	287.737872
拡張倍精度	2047.347488	4682.559193

case2　　実行時間(秒)

精度	HD5880		Phi5110P	
	1面	2面	192core	240core
倍精度	27.190329	13.814706	19.255160	17.815714
拡張倍精度			302.454400	256.691088
4倍精度	453.623999	232.079049		

Phi5110Pは倍精度は高速でE5-2670は拡張倍精度が高速。HD5880の4倍精度はdd形式のため高速です。

5.4　5次元積分計算

5次元積分実行結果一覧表							
演算量	(N=120)						
case	GFLOP						
case1	2514						
case2	2638						
実行時間一覧表(秒)							
CPU	SR16000		E5-2670		Phi5110P		HD5880
case	32core	64smp	c	fortran	c	fortran	2面
case1	8.404	5.426	27.756	27.621	7.474	7.665	5.868
case2	9.841	6.373	28.275	27.921	7.324	7.462	5.928

演算量が約2500GFLOP程度だとSR16000とPhi5110P,HD5880の性能はほぼ等しい.
これは4次元積分でも同様でした。

5.5 6次元積分計算

6次元積分実行結果一覧表	
演算量	(N=120)

case	GFLOP
case1	312064
case2	243328
case3	232896

実行時間一覧表(秒)

cpu	SR16000		E5-2670		Phi5110P		HD5880
case	32core	64smp	c	fortran	c	fortran	二面
case1	2223.744	1418.553	2385.352	2907.167	962.356	943.418	589.471
case2	1929.049	1256.823	2096.437	2717.488	925.742	959.321	568.483
case3	2212.909	1408.712	2467.079	4071.100	888.431	1063.900	606.803

演算量がこれだけ大きくなると,Phi5110P,HD5880の性能がSR16000を大きく上回ります。また,Phi5110P,HD5880のカタログ性能はほぼ同じですがHD5880の性能がPhi5110Pの倍近く良くなっています。これはPhi5110Pは60CPU,240coreという構成によるものと考えられます。