数値計算と性能,精度に関するまとめ

1.はじめに

数値計算の性能,精度に関して以下の事をまとめました。

(1) SR16000のsmpとmpiの効果を対称行列のノンゼロ要素のみを使用するCG法でその性能を検証しました。

(2) ループ積分を用いて,アクセラレータ及びサーバーの性能を比較しました。

(3) サーバーにおいてRump's例題でDD形式とIEEE形式の性能を比較しました。

(4) サーバーにおいて3次元ループ積分でDD形式とIEEE形式の性能を比較しました。

尚,チューニングの方法や精度に関する注意事項は適宜記載しています。

2 SR16000でのsmpとmpiに関して

SR16000では、1coreに関して2組の乗加算器があり一般的にsimd命令が適用される場合はsmt=off,適用されない場合はsmt=onの性能がよくなります。1ノードでの実行では以下の様になります。

Simd命令が適用されない

```
 setenv HF_BINDPROC_STRIDE 1
./cg_omp -F'prunst(threadnum(64),BIND(0))'
```

Simd命令が適用される

```
 setenv HF_BINDPROC_STRIDE 2
./cg_omp -F'prunst(threadnum(32),BIND(1))'
```

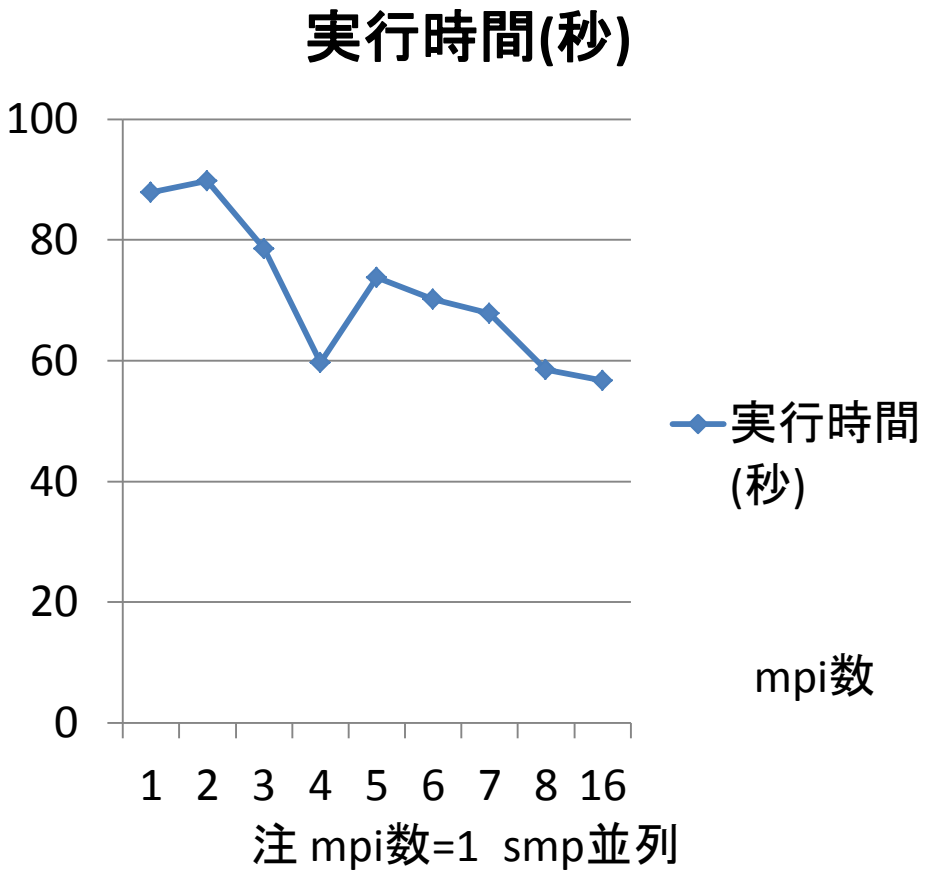
またSR16000の構成では最大8ノードで,
1ノード32coreのため,1ノード当たりのコア数がノード数より多いのでハイブリッド並列の効果に特徴があります。参考までにBG/Lは1ノード2core,BG/Qは1ノード16coreとノード数が1ノード当たりのコア数より多くなります。

1ノードでのmpi数とsmp数

Simd命令が適用されるのでmpi数*smp数=32

node =1
total_tasks = 16
poe mpibind -thread 2 -hpc ./cg_mpi5

SR16000　cg法
N = 2016000
スレッド数 = int(32 / mpi数)
1mpi,1スレッド1123.0154秒



1,4,8ノードでの性能

SR16000　cg法

N＝10000000

1node,1mpi,1スレッド11547.3647秒

case1　1node,2mpi,16smp

case2　1node,4mpi,8smp

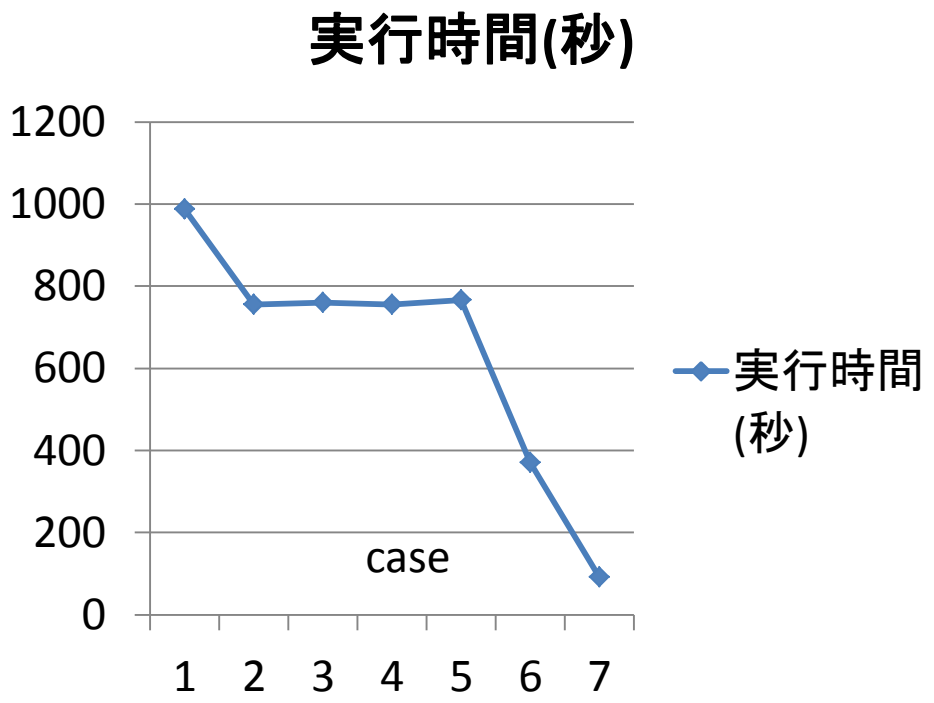
case3　1node,8mpi,4smp

case4　1node,16mpi,2smp

case5　1node,32mpi,1smp

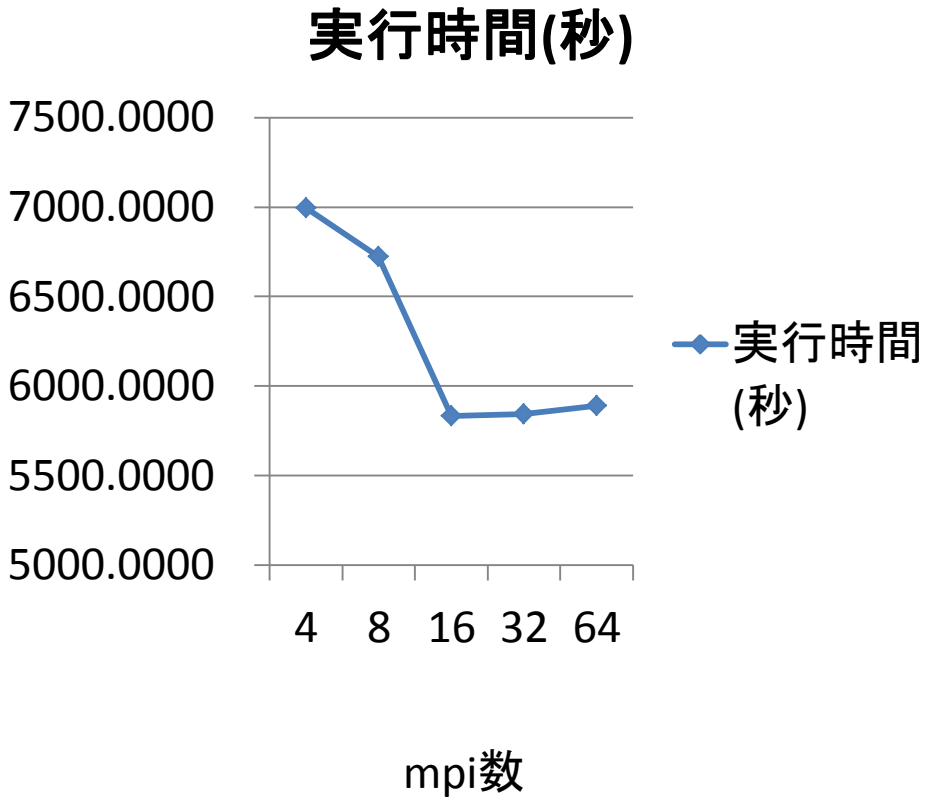
case6　4node,4mpi,32smp

case7　8node,8mpi,32smp



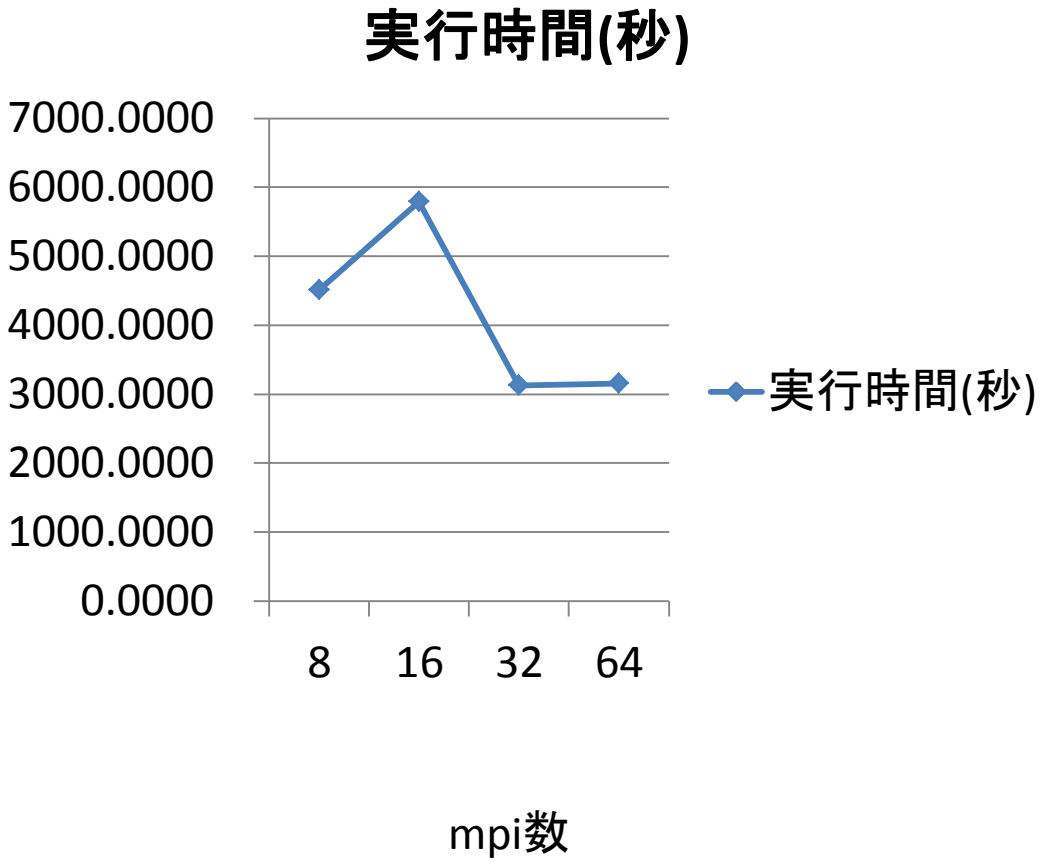
SR16000　4ノード
N=100000000
Mpi数*smp数=128

node =4
 total_tasks =16
poe mpibind -thread 8 -hpc ./cg_mpi5

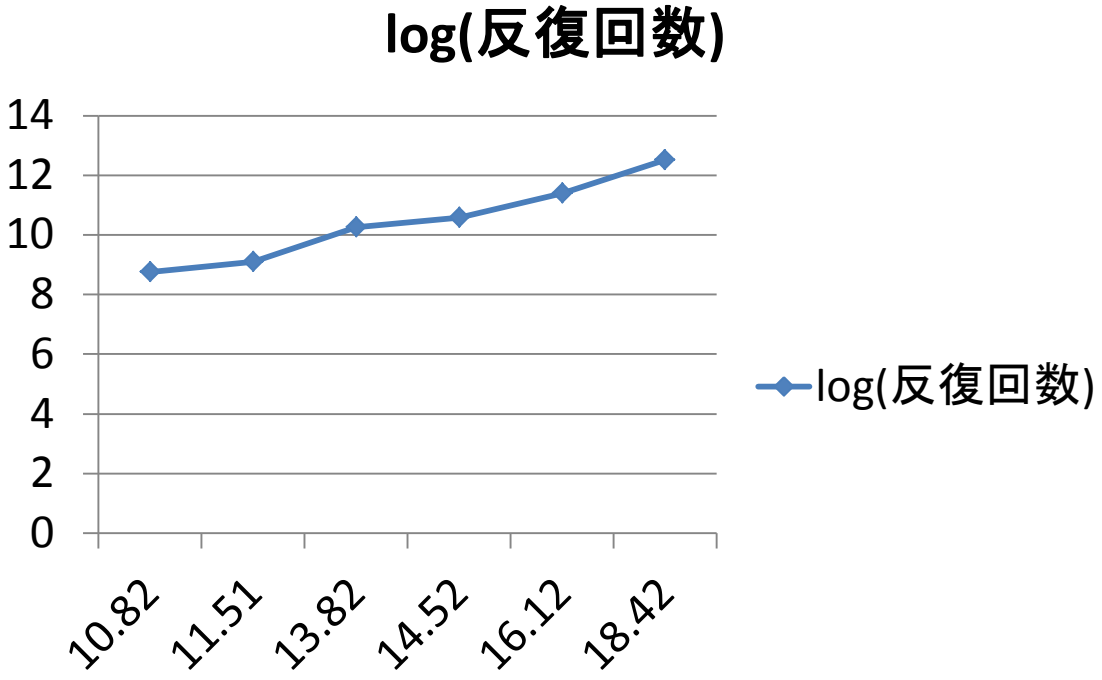


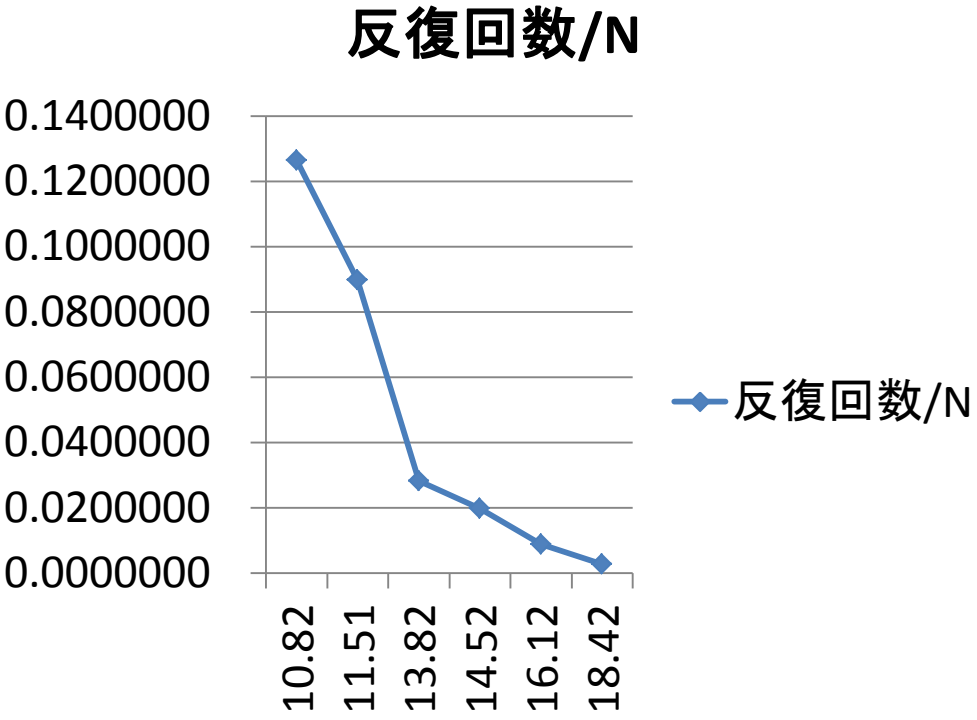
SR16000　8ノード
N=100000000
Mpi数*smp数=256

```
node =8
 total_tasks =32
poe mpibind  -thread  8 -hpc ./cg_mpi5
```



サイズNに対する反復回数
サイスNが大きくなると反復回数は増加する。
サイズNが大きくなると反復回数/Nは小さくなる。





高エネルギー加速器研究機構のSR16000の
構成は1ノードあたり32coreとなっていますが、
32coreでメモリを蜜結合で共有するのではなく,
4つの8coreで蜜結合が疎結合でメモリを共有する
形となっています。このため、
1ノード32coreではなく,4ノード8coreと考えた方
が良く1ノードでは32smp並列より,
4mpi,8smpのハイブリッド並列の方が性能が
でます。4ノード,8ノードの場合も同様に
8core16ノード,32ノード考えて実行すると
効果がでます。
またsimd命令が適用できないプログラムでは
Smt=onとして,2smp/coreとして実行する
と性能がでます。

3. ループ積分を用いて,アクセラレータ及びサーバーの性能比較

次の3つのループ積分を用いてアクセラレータ及びサーバーの性能比較をしています。

4次元積分　S221

$$S^{221}(s;m_1^2,m_2^2,m_3^2,m_4^2,m_5^2)=\int_0^1\int_0^{1-x}\int_0^{1-x-y}\int_0^{1-x-y-z}\frac{1}{DC}dudzdydx$$

$$C=(x+y+z+u)(1-x-y-z-u)+(x+y)(z+u)$$
$$E=(1-x-y-z-u)(x+z)(y+u)+(x+y)zu+(z+u)xy$$
$$M^2=xm_1^2+ym_2^2+zm_3^2+um_4^2+(1-x-y-z-u)m_5^2$$
$$D=-sE+M^2C$$

$$s^{221}(-1:100,100,0,0,100)$$

解析近似解：0.038004438127

6次元積分 Laporta Ⅰ

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

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

```
C=(x1+x2+x3+x4+x5)*(x1+x2+x3+x6+x7)-(x1+x2+x3)**2

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)
```

```
m12=1.0d0, m22=1.0d0, m32=1.0d0,m42=1.0d0
m52=1.0d0, m62=1.0d0, m72=1.0d0
p12=1.0d0, p22=1.0d0, p32=1.0d0,p42=1.0d0
s =1.0d0
t =1.0d0
```

解析近似解=0.0853513981538

8次元積分M44

$$I = \int_{\Omega} \frac{1}{C(D - i\varepsilon C)} d\Omega$$

$$\Omega = \{(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) \mid$$
$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0,$$
$$x_5 \geq 0, x_6 \geq 0, x_7 \geq 0, x_8 \geq 0, x_9 \geq 0,$$
$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = 1\}$$

$$x_{i,j,k,\ldots} = x_i + x_j + x_k + \ldots..$$

$$C = x_{167}x_{278}x_{349}x_{589} - x_{349}x_{589}x_7{}^2 - x_{167}x_{349}x_8{}^2$$
$$- x_{167}x_{278}x_9{}^2 + x_7{}^2x_9{}^2$$

$$D = -x_{167}x_{278}x_3{}^2x_{589} - x_{167}x_2{}^2x_{349}x_{589} - x_1{}^2x_{278}x_{349}x_{589}$$
$$+ x_{123}x_{167}x_{278}x_{349}x_{589} - 2x_1x_2x_{349}x_{589}x_7 + x_3{}^2x_{589}x_7{}^2$$
$$- x_{123}x_{349}x_{589}x_7{}^2 + x_{167}x_3{}^2x_8{}^2 + x_1{}^2x_{349}x_8{}^2$$
$$- x_{123}x_{167}x_{349}x_8{}^2 - 2x_{167}x_2x_3x_8x_9 - 2x_1x_3x_7x_8x_9$$
$$+ x_{167}x_2{}^2x_9{}^2 + x_1{}^2x_{278}x_9{}^2 - x_{123}x_{167}x_{278}x_9{}^2$$
$$+ 2x_1x_2x_7x_9{}^2 + x_{123}x_7{}^2x_9{}^2$$

解析解 $= \frac{441\zeta_7}{8} = 55.58525391567 84$

8次元積分M_{44}実行結果一覧表

$N=16$, 演算量 $=850.4$GFLOP

実行時間(秒),性能(GFLOPs)一覧表

CPU	smp数	実行時間	性能	備考
Pezy	8192	3.222556	264	HPC152
	8192	2.438170	349	H27末
	8192	1.298701	655	最終
Phi5110P	240	1.838314	463	−O2
E5-2670	16	6.325043	134	−O2
HD5870	3200	1.542792	551	標準
gmp	16	9.078043	94	−O1
	16	7.021060	121	−O2
g9mpx	16	10.650697	80	−O1
	16	8.946464	95	−O2
g9mpx3	40	4.800980	177	−O1
	40	3.311994	257	−O2

Pezy	1500GFLOPs	E5-2670	2.60GHz
Phi5110P	1011GFLOPs		
HD5870	1088GFLOPs	性能緒元一覧	
gmp	E5-2690(SB) 2.90GHz		
g9mpx	E5-2687W v2 3.40GHz		
g9mpx3	E5-2687W v3 3.10GHz		

アクセラレータによる性能比較

	ループ積分性能比較		実行時間(秒)一覧表		
プログラム	N	演算量	Pezy-SC	Phi5110P	HD5870
		(GFLOP)	1500GFLOPs	1011GFLOPs	1088GFLOPs
s221	512	3711	6.911009	14.119193	6.374162
s221	1024	59374	110.521333	194.452293	135.027731
laporta i	64	10308	16.988779	20.838768	14.426863
laporta i	128	659707	930.784595	1515.707021	904.091546
m44	16	850	1.298701	1.838314	1.542792
m44	32	217703	331.900691	298.712631	330.427198

発生した問題点

1.m44 HD5870　N=32 でゼロ割発生

2. m44　Pezy-SC 除算の性能が悪い。

（＜＝今後改善される模様）

対策

HD5870　N=16,Pezy-SC では除算が遅いが結果は正しい計算をしている事に着目

=>二分法の考え方を使用する。

HD5870のN=32での対策

M_{44}の計算を模式化すると$\int_{\Omega}\frac{1}{CD}d\Omega=\sum\frac{W}{CD}$の計算をしている。二重 指数関数型積分では

$0<CD<<1$では$0<W<CD<<1$となっている。

$\varepsilon=CD$として,$\sum\frac{W}{\varepsilon}$の計算で桁落ちなどにより

ゼロ割が発生する。このため,ε_0を定め,$\varepsilon<\varepsilon_0$

なら$\varepsilon=\varepsilon_0$として計算する。最終 次元での式は

$$\int_0^1\frac{(\log(x))^6}{1-x}dx=\sum x(\log(x))^6\sqrt{(\log(\frac{1-x}{x})^2+\pi^2}$$

となり,$x=e^{-7}$,$W_M=\sum x(\log(x))^6\sqrt{(\log(\frac{1-x}{x})^2+\pi^2}$

≈ 823

$\varepsilon_0=CD\approx x^2$とすると,

$\varepsilon_0=10^{-40}$,$x=10^{-20}$,$W_\varepsilon=4.40\times10^{-9}$

$\varepsilon_0=10^{-50}$,$x=10^{-25}$,$W_\varepsilon=2.10\times10^{-13}$

より,$\varepsilon_0=10^{-50}$とすれば倍精度演算の 範囲では最終結果に影響がなく,値はε_0の減少関数で、$\varepsilon_0->0$で桁落ちで∞となるので二分法でε_0の値を定める事が出来る。

Pezy-SCの除算の高速化
(今後はこの修正が必要なくなる)

現時点では$\mathrm{Pezy-SC}$では単精度の$\frac{1}{\sqrt{x}}$しか高速に実行されない。$(x > 2^{-149} \approx 1.4\times10^{-45})$

このため$\varepsilon_0 = 10^{-40}$とする。

$\varepsilon < 1.0\times10^{-40}$に対し,$\frac{W}{\varepsilon} = \frac{Wh_0}{\varepsilon_0}$となる共通の$h_0$を求める。$h_0 = 1$なら$\frac{W}{\varepsilon} > \frac{W}{\varepsilon_0}, h_0 >> 1$とすれば

$\frac{W}{\varepsilon} < \frac{Wh_0}{\varepsilon_0}$となるので2分法で,$\frac{W}{\varepsilon} = \frac{Wh_0}{\varepsilon_0}$となる$h_0$を求める。コーデイン グでは

```
h = 1.0e00;                    f, g, h : double    ε : float
if(g >= ε0){ε = g;}
else   {ε = ε0; h = h0}
f = rsqrt(ε);
W/ε = W * f * f * h;
```

fの精度不足分はh_0に反映させて積分値全 体の精度低下を抑える。

M44　修正結果と結果の精度への影響

解析解=55.5852539156784

HD5870

N=16　　55.5852539156735

N=32　　55.5852539156776

N=32で修正した結果は以前と同じ精度

Pezy-SC

N=16　　55.5852539156731

単精度演算を使用しても必要な精度を保っています。

4. Rump's例題を用いたサーバーでのDD形式とIEEE形式の性能比較

rump's例題とは,四則演算からなる簡単な計算で,有効ビット数が121ビット以上を要求する計算のため,精度、性能の検証によく使用されるものです。

$$a = 77617.0, b = 33096.0$$

$$f = 333.75b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + 5.5b^8 + \frac{a}{2b}$$

$$a^2 = 5.5b^2 + 1 = 6024398689$$

$$a^2(11a^2b^2 - b^6 - 121b^4 - 2) = -5.5b^2 - 333.75b^6 - 2$$

より, $f = -2 + \frac{a}{2b} = -\frac{54767}{66192} = -0.827396059946$

DD形式
変数の値を複数個の倍精度変数の和で
表す方式。
演算順序の変更がないように注意が必要。
サーバー系では-fp-model　preciseが必要
n個の倍精度変数の和を2n倍精度という。
nが大きくなると,演算量は乗算がnの3乗,
加減算がnの2乗で増える。
またnが3以上となると除算で途中の結果で
絶対値が大きい順に並べる注意が必要と
なる。

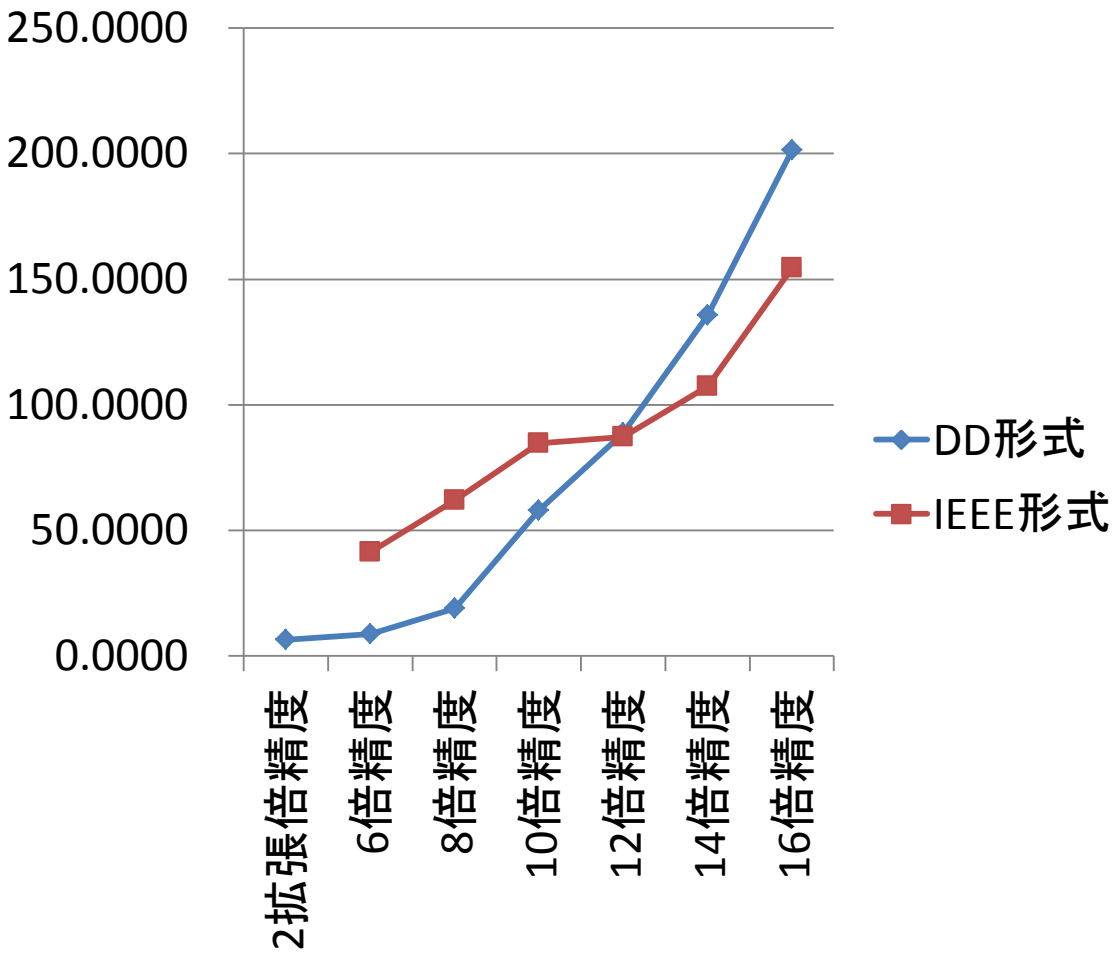
IEEE形式
IEEE754-2008での4倍精度の仮数部を拡張する。
符号部1ビット,指数部15ビット,仮数部32*p-16ビット
をp倍精度という。演算がコール形式となるためpが
小さい場合、オーバーヘッドが大きくなる。コーデイング
では乗算、除算は比較的容易で、加減算での桁落ち
数の計算で誤りが生じ易い。

サーバー系では80ビットの内部レジスタを有するものが多く
拡張倍精度や、複数個の拡張倍精度の和の演算が有効
になる。特にオプション-fp-model　extended は
性能に効果的な場合が多い。

Rump's 例題　N=500000000
E5-2687W v3 3.10GHz　40smp

拡張倍精度+拡張倍精度　有効ビット数　130>121
10倍精度から12倍精度で実行時間が逆転

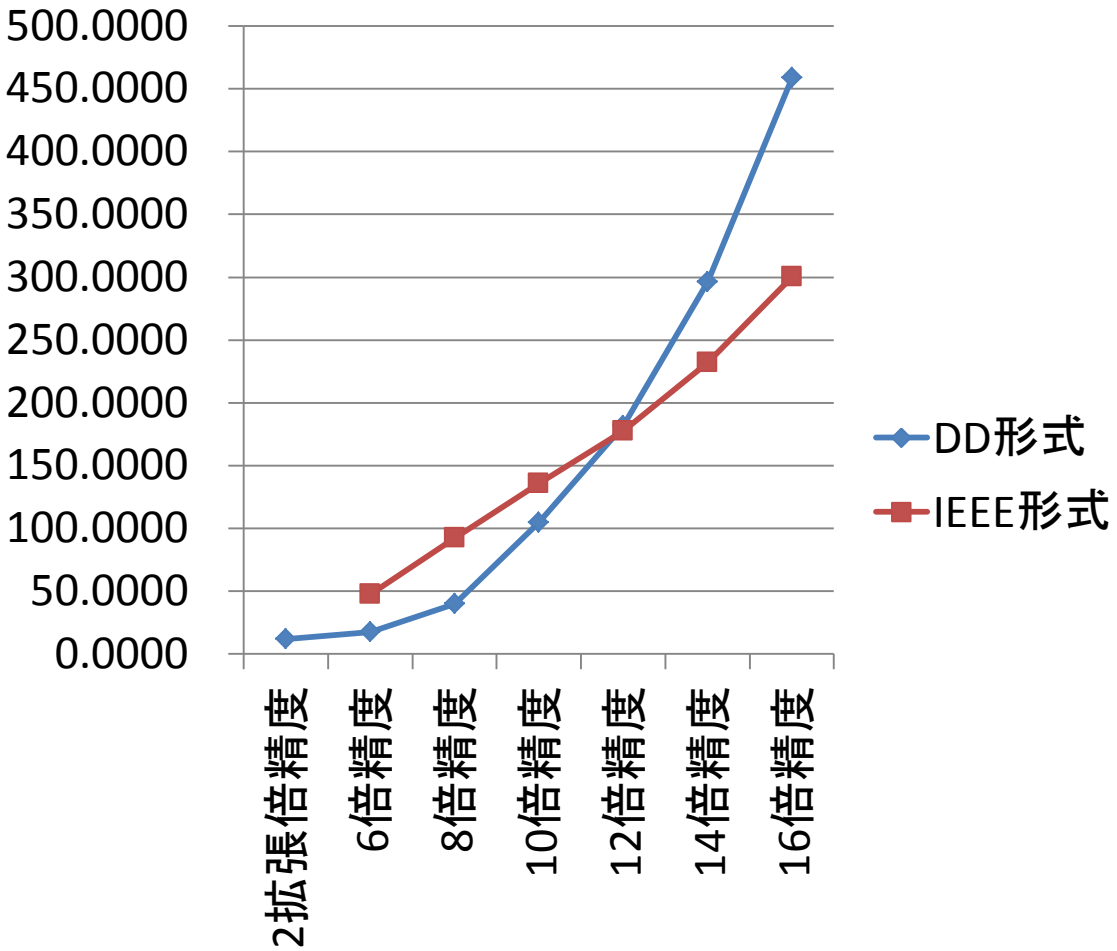
実行時間(秒)



Rump's 例題　N=500000000
E5-2687W v2 3.40GHz　16smp

拡張倍精度+拡張倍精度　有効ビット数　130>121
10倍精度から12倍精度で実行時間が逆転

実行時間（秒）



5. 3次元ループ積分でのサーバーでの DD形式とIEEE形式の性能比較

使用した3次元積分は以下のinfra boxと言われるものでこの問題では10進6桁正しい結果を得るには

$x = 1 - \varepsilon$で$\varepsilon \le 10^{-32.8} = 2^{-108}$のxが取れる必要がある。

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

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

$$s = -500^2, t = -150^2, m_f = 150, m_e = 0.0005, \lambda = 10^{-30}$$

解析近似解 $= \dfrac{1}{-s(-t+{m_f}^2)} \ln(\dfrac{-s}{\lambda^2}) \ln \dfrac{(-t+{m_f}^2)^2}{{m_f}^2 {m_e}^2}$

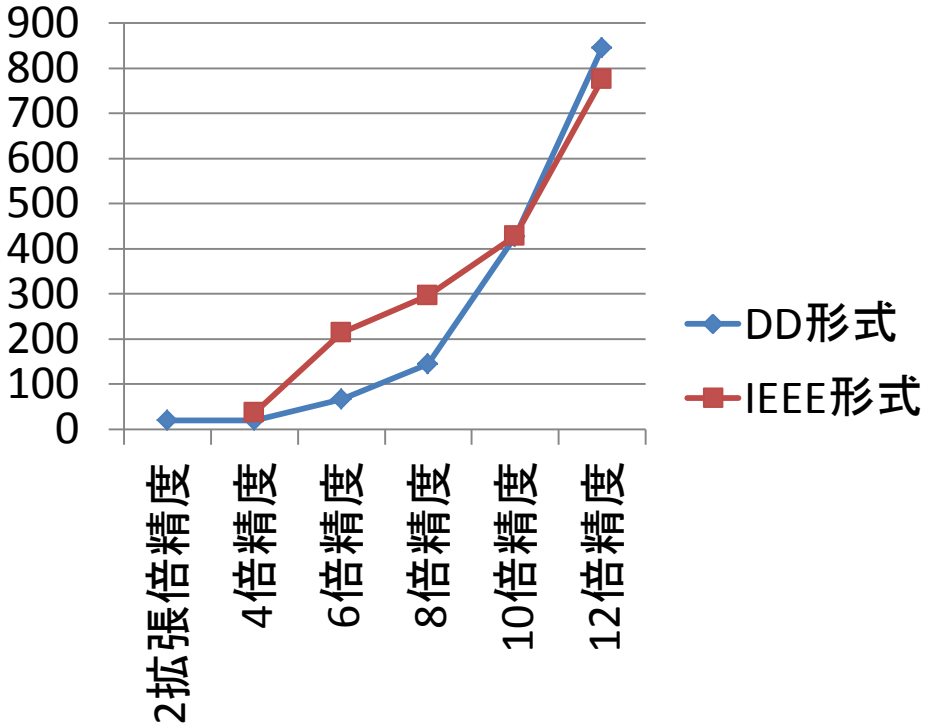
相対誤差 $= \dfrac{\lambda}{\ln(\dfrac{-s}{\lambda^2})} \dfrac{\pi(-t+{m_f}^2)}{m_e \ln \dfrac{(-t+{m_f}^2)^2}{{m_e}^2 {m_f}^2}} = 7 \times 10^{-26}$ 以下

Infra box
E5-2690(SB) 2.90GHz 16smp
N=1024

DD形式4倍精度のみa,b,c
X=a+b をx=1-c (a,b,c>0) と表す
コーデイング。

10倍精度で実行時間がほぼ同じ

実行時間(秒)

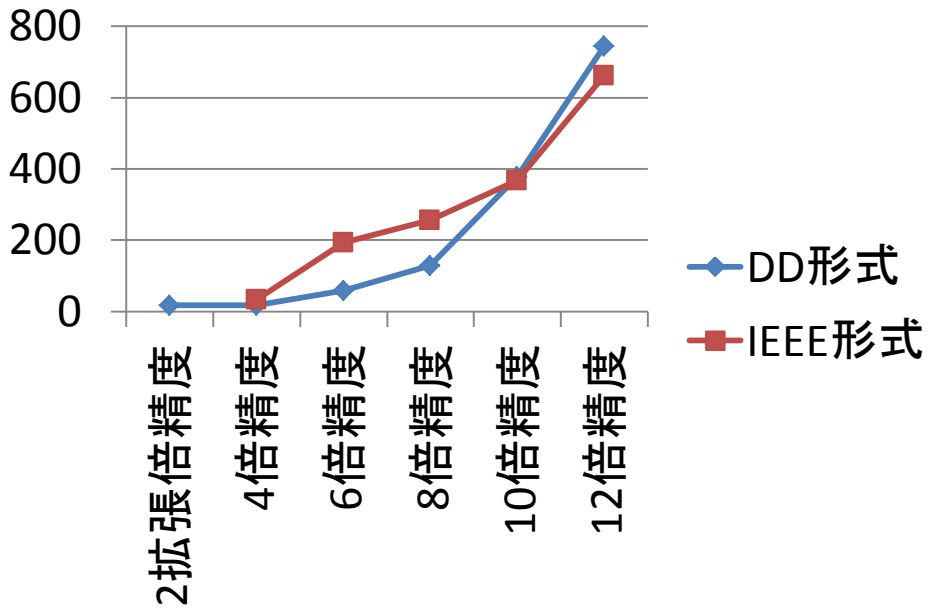


Infra box
E5-2687W v2 3.40GHz 16smp
N=1024

DD形式4倍精度のみa,b,c
X=a+b をx=1-c (a,b,c>0) と表す
コーデイング。

8倍精度から10倍精度で実行時間が逆転

実行時間(秒)



DD形式4倍精度 (a,b,c>0)
X=a+b

```
 do i=1,n
x=-1024+i
t=x*h
y=pi*0.5q0*sinh (t)
x30 (i) =exp (y) / (exp (y) +exp (-y) )
gw30 (i) =cosh (t) *pi/ (exp (y) +exp (-y) ) **2
end do
```

コーデイングでは赤字の1行変更するだけで良い。

X=1-c

```
 do i=1,n
 x=-1024+i
 t=x*h
 y=pi*0.5q0*sinh (t)
 x30 (i) =1.0q0/ (1.0q0+exp (-2.0q0*y) )
gw30 (i) =cosh (t) *pi/ (exp (y) +exp (-y) ) **2
 end do
```