各種計算機アプリケーション性能比較

目次

各種計算機

アーキテクチャの相違は性能のみならず,精度,コンパイラの最適化機能,互換性にも影響が出てきます。

使用した計算機は以下の5つです。

(ア)SR16000/M1

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

(イ)SR16000/XM1

SR16000/xm1は周波数が3.3GHzで他はSR16000/M1 と同じです。
演算性能だけみれば,SR16000/M1 1ノードはSR16000/xm1の16%性能向上版ともいえます。

(ウ)T2K 筑波システム　AMD quad-core Opteron 8000 シリーズ
(Barcelona)
1 node:ピーク性能 147GFLOPs,16MPI/node

(エ)GPGPU
GPU カード型番:ATI RadeonHD5880
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

(オ)　SR11000/K1(1ノード)
プロセッサ:power5
周波数:2.1GHz
CPUコア数 16
論理コア数 16
理論最大性能　134.4 GFLOPs
メモリ容量 32GB
メモリアーキテクチャー　Flat Memory　Interleave
L3キャッシュ　Off-Chip 36MB/2コア
演算器/物理コア　乗加算器2つ

記述に関してはそれぞれ
(ウ)　T2k
(エ)　GPGPU
(オ）　SR11000
としています。

1. 多倍長演算アプリケーション

測定した数値積分の積分式,解析近似解は以下の様なものです。

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

式

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

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

データ

解析近似解

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

相対誤差7×10^{-26}以下

ここで,変数の内容と単位は以下の様になっています。

m_e :電子の質量(GeV), m_f :フェルミ粒子の質量(GeV)

s,t :衝突エネルギー(GeV^2), $t<0$

λ :仮想光子質量(GeV),実際の物理計算では10^{-30}あたり。

(注)$1Gev = 10^9 eV$, $1eV = 1.602\times10^{-19} J$

実験との対応で要求される積分結果の精度は
相対誤差0.1%以下です。

この積分計算は積分領域内で特異点（Ｄ＝０）が
存在（赤外発散）。

＝＞微小量εを使用して, $D->D-i\varepsilon$として有理化し
発散を回避(正確には$s->s+i\varepsilon$)
＝＞$\varepsilon->0$の極限の値を得てそれを目的とする積分値
としています。

測定では,数値積分法と加速法は以下のものを使用しています。

数値積分法:二重指数関数型積分法
加速法: $\varepsilon-$算法(*Wynn*のアルゴリズム)

　実数部の積分計算結果が要求される相対誤差0.1%以下をみたすのに要する時間は64スレッド（ＭＰＩ）で以下の様に変遷しています。

実行時間一覧表(秒)

CPU	SR11000	SR16000/XM1	T2K	SR16000/M1
理論性能(GFLOPs)	537.6	844.8	588	980.49
4倍精度	681	589	414	224
6倍精度	346	401	372	208
8倍精度	382	458	708	314

多倍長精度演算結果（実数部）

解析近似解　-0.3561736812D-06

4倍精度　-0.3560594322D-06　相対誤差　0.032%

6倍精度　-0.3561608223D-06　相対誤差　0.004%

8倍精度　-0.3561449971D-06　相対誤差　0.008%

2　倍精度演算アプリケーション

倍精度演算で十分な精度がでて並列化効果の大きい
4つの数値積分プログラムで性能を比較しています。

(a)s221

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

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

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

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

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

テストケース

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

解析近似解$:0.0380004438127$

(b)laportad

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

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

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

解析近似解=0.2762092253588

(c) laportag

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

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

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)

解析近似解=0.1723367907503

(d)laportah

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

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

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*x
4*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)

テストデータ

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=0.1036407209893

実行結果の一覧を示します。

演算量はソースプログラム上でカウント(最適化によるコンパイル前)していますので,性能は良い方向にでる傾向がありますので,目安としてください。

結果の検証

解析近似解

s221 = 0.0380004438129
laportad = 0.2762092253590
laportag = 0.1723367907503
laportah = 0.1036407209893

T2K

s221 = 0.0380004438127
laportad = 0.2762092253590
laportag = 0.1723367907502
laportah = 0.1036407209892

SR16000/XM1

s221 = 0.0380004438127
laportad = 0.2762092253590
laportag = 0.1723367907502
laportah = 0.1036407209892

GPGPU

s221 = 0.0380004438126
laportad = 0.2762092253588
laportag = 0.1723367907501
laportah = 0.1036407209891

SR16000/M1

s221 =0.0380004438127
laportad=0.2762092253469
laportag=0.1723367907503
laportah=0.1036407209893

T2K

プログラム	次元数	演算量	64MPI	128MPI	256MPI
		(TFLOP)	(秒)	（秒）	（秒）
s221	4	6.27429	31.807	15.888	8.007
laporatd	5	3.981312	11.927	5.959	2.967
laportag	6	391.1639	1257.935	621.679	311.828
laportah	6	394.1499	1151.404	580.99	290.99
性能（ピーク） 147GFLOPs/node					
	4node	0.588TFLOPs			
	8node	1.176TFLOPs			
	16node	2.352TFLOPs			

SR16000/xm1

プログラム	次元数	演算量	16MPI	32MPI	64MPI
		(TFLOP)	(秒）	（秒）	（秒）
s221	4	6.27429	63.698	39.823	29.902
laporatd	5	3.981312	21.75	12.998	8.102
laportag	6	391.1639	3408.677	1956.102	1280.051
laportah	6	394.1499	2139.455	1500.685	1194.661
プログラム	次元数	演算量	16スレッド	32スレッド	64スレッド
		(TFLOP)	(秒）	（秒）	（秒）
s221	4	6.27429	36.236	28.769	15.257
laporatd	5	3.981312	19.543	12.472	6.533
laportag	6	391.1639	3017.369	1768.963	1162.71
laportah	6	394.1499	1609.981	861.16	793.411
	理論ピーク性能 0.844TFLOPs				

1ノードの場合はSMP並列がMPI並列より性能が上回っています。

GPGPU

プログラム	次元数	演算量	gpu0	gpu1	gpu0,1
		(TFLOP)	(秒)	(秒)	(秒)
s221	4	6.27429	27.19	27.191	13.815
laporatd	5	3.981312	11.257	11.251	5.85
laportag	6	391.1639	1176.892	1176.6	589.553
laportah	6	394.1499	1132.186	1130.624	569.934
	理論ピーク性能				
		gpu0,gpu1	0.544TFLOPs		
		gpu0,1	1.088TFLOPs		

(注)1600コアGPU0,GPU1 3200コア GPU0,1

SR16000/M1

プログラム	次元数	演算量	16MPI	32MPI	64MPI
		(TFLOP)	(秒)	(秒)	(秒)
s221	4	6.27429	28.689	16.008	12.344
laporatd	5	3.981312	18.675	11.099	6.312
laportag	6	391.1639	2754.579	1520.859	1048.324
laportah	6	394.1499	1639.434	977.612	954.2
プログラム	次元数	演算量	16スレッド	32スレッド	64スレッド
		(TFLOP)	(秒)	(秒)	(秒)
s221	4	6.27429	27.265	14.429	11.774
laporatd	5	3.981312	16.838	8.687	5.523
laportag	6	391.1639	2575.904	1293.523	985.318
laportah	6	394.1499	1355.26	687.146	617.921

Xm1と同じく1ノードの場合はSMP並列がMPI並列より性能が上回っています。