各種計算機アプリケーション性能比較

平成２５年度第3四半期

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。主に使用した計算機は以下の2つです。

(ア)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

(イ)BG/Q

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

これらのほかに

- x5570 8コア　2.93GHz キャシュ　8MB/コア
- e5430 16コア　2.66GHz キャシュ 12MB/コア
- T2kシステム

などとも比較しています.

1.はじめに

今回は主に量子モンテカルロ法による
物性スペクトル計算で使用したDD形式の演算と
ieee754-2008形式の整数演算の演算量と
実測値との関係に関して記述しています。
比較は,SR16000/M1,BG/QでのDD方式の結果
と,整数演算でのX5570,E5430,T2Kシステム
の結果に関して行っています。
また演算量の検証として,X5570,e5430シングル
ジョブでDD形式の場合とieee754-2008形式
での場合の比較を行っています。この結果では
アルゴリズムを同じ(ソースチューニングなし)の
場合,8倍精度演算ではどちらの方式もほぼ同じ
性能で,10倍精度演算ではieee754-2008形式
(整数演算方式)の場合の性能が良く、
浮動小数点演算性能のみでプログラムの性能を
比較する方式の問題点を示しています。

2.DD形式方式と整数演算方式の演算量

2.1　DD形式の$2n$倍精度の演算量

$m=[\ln_2 n]+1$

(1)$2n$倍加減算

基本精度加減算　$9n^2-14n+6$

(2)$2n$倍精度乗算

基本精度加減算　$2(n-1)(n^2+4n-3)$

基本精度乗算　$\dfrac{n(7n-5)}{2}$

(3)$2n$倍精度除算

基本精度加減算　$(n-1)(n^3+15n^2-22n+6)$

基本精度乗算　$\dfrac{n(n-1)(7n-5)}{2}$

基本精度除算　n

(4)$2n$倍精度平方根(除算方式)

基本精度加減算　$m(n-1)(n^3+15n^2-22n+6)$
$+(m+1)(9n^2-14n+6)$

基本精度乗算　$\dfrac{mn(n-1)(7n-5)}{2}$

基本精度除算　mn

基本精度平方根　1

(5)$2n$倍精度平方根(逆数平方根方式)

基本精度加減算　$2(3m+1)(n-1)(n^2+4n-3)$

2.2 整数演算方式の演算量

整数演算方式では, 有効ビット(仮数部のビット数 $+1$)を30ビット単位で操作しています。$P(>4)$倍精度演算の有効ビット数m

$m = 32 \times P - 16 + 1$となります。このため実行時間は

$n = \left[\frac{m-1}{30} \right] + 1$の関数となります。通常演算量の単位は

浮動小数点演算のため, 整数演算方式では, 異なる精度での実行時間比を見積もる指標として, 演算量はnの関数形で定めています。ここで, k, k_{add}, k_{mult}は比例定数を表しています。

(1)加減算

$k_{add} \times n$

(2)乗算

$k_{mult} \times n^2$

(3)除算

$2 \times k_{mult} \times n^2$

$\frac{a}{b}$は$a \times \frac{1}{b}$で計算し, $\frac{1}{b}$の計算は反復法ではなく直接整数演算で求めていてほぼ乗算と同じ実行時間になっています。

(4)平方根

$2lk_{mult} \times n^2 + (l+1) \times k_{add} \times 4$倍精度平方根

$l = \left[\ln_2 \frac{n}{4} \right] + 1$, 除算$l$回, 加減算$l+1$回より。

簡単に$k \times n^{2.5}$と見積もる場合もあります。

3.量子モンテカルロ法による物性スペクトル計算

3.1　6,8,10倍精度演算での性能比較

DD形式と整数演算方式の比較

測定ソース		
	DD形式	SR16000/M1でもっともチューニングされたもの 6倍精度演算は今回新たにチューニングされたもの
	整数演算方式	
		オリジナルソースの演算部分を多倍長精度演算 に置き換えたもの

指定オプション

DD形式　$-O3 \quad -fp-\mathrm{mod}\,el \quad strict$

整数演算方式　**標準**$(-O2)$

測定条件

6倍精度　$N=100, L=448, \beta=10, U=6$

8倍精度　$N=100, L=448, \beta=20, U=8$

10,12**倍精度**　$N=100, L=448, \beta=20, U=10$

実行時間一覧表(秒)

CPU	X5570		E5430	
	整数演算	DD形式	整数演算	DD形式
6倍精度	1013.001	496.412	1368.826	696.412
8倍精度	1471.272	992.806	1913.822	1338.291
10倍精度	1997.684	2314.178	2693.822	3108.519
12倍精度	2931.289		3945.040	

整数演算方式では,P=5,6,7ではn=p,
P=8,10,12ではn=p+1となるため実行時間比率で見ると8倍精度演算が大きくなります。

DD形式の8倍精度と10倍精度の実行時間の比

CPU	比率
SR16000	2.15
BG/Q	2.30
x5570	2.33
e5430	2.33
T2K	2.29

演算量

8倍精度

演算	加減算	乗算	除算	平方根	計
加減算	94				94
乗算	174	46			220
除算	858	138	4		1000
平方根	2950	414	12	1	3377

10倍精度

演算	加減算	乗算	除算	平方根	計	比率
加減算	161				161	1.71
乗算	336	96			432	1.96
除算	2084	384	5		2473	2.47
平方根	6896	1152	15	1	8064	2.39

整数演算方式

対8倍精度実行時間比

精度	x5570	e5430	計算
10倍精度	1.358	1.408	1.436
12倍精度	1.992	2.061	1.942

計算

10倍精度 $(\frac{304}{240}+(\frac{304}{240})^2)\div 2=1.436$

12倍精度 $(\frac{368}{240}+(\frac{368}{240})^2)\div 2=1.942$

DD形式か整数演算方式か？

分岐点は8倍精度にある模様

原因

(1) DD形式では8倍精度,10倍精度となると倍精度加減算の比率が多くなるため,乗加算命令の効果がなくなる。

(2) DD形式では演算量から,最適化がうまく効くのは8倍精度までで,10倍精度以上では最適化の効果が低下する。

(3) 整数演算方式では4倍精度演算から16倍精度演算の間では、他の演算精度と比較して8倍精度演算が実行時間比率で悪くなる。

精度の面からみると,8倍精度演算では

DD形式では$N=100, L=448, \beta=20, U=8$

整数演算方式では$N=100, L=448, \beta=20, U=10$

までと,10進10桁一致する解が得られる領域に差が,出ている。

またDD形式では更に精度をあげても,表現できる数値範囲の問題で10倍精度演算,$N=100, L=448, \beta=27, U=10$までしか計算できない。

3.2 x5570,e5430でのDD形式と ieee754-2008形式(整数演算方式)の比較

DD形式と整数演算方式の比較

測定ソース

オリジナルソースの演算部分を多倍長精度演算に置き換えたもの

指定オプション

DD形式 $-O3 \quad -fp-\mathrm{mod}\,el \quad strict$

整数演算方式 標準$(-O2)$

測定条件

5,6,7倍精度 $N=100, L=448, \beta=10, U=6$

8倍精度 $N=100, L=448, \beta=20, U=8$

10,12倍精度 $N=100, L=448, \beta=20, U=10$

性能比較一覧表

実行時間(秒)一覧表

演算精度	x5570		e5430	
方式	整数演算	DD方式	整数演算	DD方式
5倍精度	896.776		1204.938	
6倍精度	1013.001	752.237	1368.826	945.859
7倍精度	1194.853		1617.087	
8倍精度	1471.272	1339.420	1913.822	1636.684
10倍精度	1997.684	3009.714	2693.822	3614.172
12倍精度	2931.289		3945.040	

有効ビット数 DD形式8倍精度

(52+1)*4=212ビット

整数演算方式7倍精度

32*7-16+1=209ビット

整数演算7倍精度	
β =20	
U	必要ビット数
7.5	187.4
7.6	188.7
7.7	189.0
7.8	191.1
7.9	192.3
8.0	193.6

7倍精度整数演算方式

$$\beta = 20, N = 100, L = 448$$

u = 7.5 - 0.1000000000D + 01

u = 7.6 - 0.9999999999D + 00

u = 7.7 - 0.9999999997D + 00

u = 7.8 - 0.1000000008D + 01

u = 7.9 - 0.1000000092D + 01

u = 8.0 - 0.9999983747D + 00

DD形式8倍精度と有効ビットの差は3ビット