

MasParを用いたエイリアス確率の計算 - 改良

4 L - 6

岩崎 一彦 吉川秀之
千葉大学工学部情報工学科

1. まえがき

超並列計算機MP-1を用いたエイリアス確率計算の改良をおこなった。スーパーコンピュータS820/80と比較する。

2. VLSIの自己テスト

VLSIのテスト法としてBISTが提案され、実用化されるようになってきた。BISTの一つの問題点はエイリアス誤りである。すなわち、テスト応答圧縮にともない、誤りが互いに打消あつて故障を見逃してしまうことである。このエイリアス確率は、2元対称チャネルに対し次式のように表される。

$$\text{Pal}(n) = 2^{-m} \sum_{j=0}^{nm} B^b(nm, J)(1-2p)^j - (1-p)^{nm}. \quad (1)$$

$B^b(nm, J)$ は、双対符号の2元重み分布を表す。 p は各ビットが誤りを含む確率、 n はテスト長、 m は圧縮回路の段数を表す[1]。計算複雑度は $O(n2^m)$ である。

3. べき計算の工夫

上記(1)の $B^b(nm, J)$ は、ビット操作および整数演算で計算できるため、MP-1における台数効果を十分生かすことができる。 $(1-2p)^j$ については倍精度浮動小数点演算となり、台数効果はあるもののMP-1の演算能力では十分でない可能性があった。 J はテスト長 n に対して単調で、かつ n が1増加したときの J の増分は高々 m であることを利用することを考えた。

● 変更前のプログラム

$\text{Pal} = \text{p_pow}(\text{plural double}(1-2*p), \text{plural double } J);$
 p_pow は、mpl言語で提供されているべき関数である。

● 変更後のプログラム

```
if (J&16!=0) Pal*=pp[16];
if (J&8!=0) Pal*=pp[8];
if (J&4!=0) Pal*=pp[4];
if (J&2!=0) Pal*=pp[2];
if (J&1!=0) Pal*=pp[1];
```

$\text{pp}[J]$ には、 $(1-2*p)^{J-1} - (1-2*p)^{J-m}$ を計算しておく。ただし、今回の問題に対して $4 \leq m \leq 16$ である。 J 、 Pal 、 $\text{pp}[J]$ はplural変数である。

4. 高速化の工夫

表1に、従来プログラム[1],[2]に対する改良項目とその効果を示す。

まず、結果の出力命令を削除した。MP-1のmpl言語では1.77倍の高速化が達成された。S820/80ではほとんど影響がなかった。

次に、上に示したようなべき計算の工夫をおこなった。この結果、4.3倍の高速化が達成された。べき計算改良の途中段階において、

```
Pal*=pp[J];
```

というプログラムを実行したところ、変更前のものと比べ遅くなってしまった。MasParはSIMDアーキテクチャなので、 $J=1, 2, \dots, m$ について、 m 回の場合分けをおこなっているようなのが原因と思われる。MasParのPEには、インデックスアドレス形式が提供されているので、アセンブラレベルのプログラムが可能ならば、さらに高速化が可能であると思われる。S820/80でも同様のべき計算の工夫をおこなったところ、2.1倍の性能改善が得られた。

並列部分の高速化が進むと、シーケンシャル部分(すなわちMP-1のPE以外が実行する部分)の計算時間が目立つようになった。そこで、2元行列の各行を整数値で表すような工夫をおこなった。MP-1で2.2倍の性能向上が得られた。S820/80では、当該部分が既にベクトル化されていたため性能改善はみられなかった。

更に、MP-1のmpl言語では、ショートインテジャ(16ビット)を使った。この結果、1.3倍の高速化が達成された。MP-1のPEは4ビットの演算器を有するので、オペランド長を短くする効果があった。

以上の工夫により、従来[1],[2]と比較して、MP-1のmpl言語では22.1倍、S820/80では2.1倍の性能向上が達成された。

改良したプログラムにおいて、時間がかかっている部分は、 m ビット整数値の全ビットの排他的論理和をおこなう部分である。各PEは

```
code=(code>>1);
code=(code>>2);
code=(code>>4);
code=(code>>8);
code&=1;
```

を実行している。 code はplural変数である。S820/80では、整数値 code に対して全ビットの排他的論理和を与えるような配列を用意した。使用したMP-1では、各PEのローカルメモリは16Kバイトなので、このような配列は使用できなかった。

5. 実行時間の比較

図1に4096PEのMP-1のmplプログラムの経過時間を示す。 $m \leq 12$ に対して、経過時間は2倍以内の差であ

る。これは、使用したMP-1が4096個のPEを持っているからである。m=14, m=16の場合、VP比は、それぞれ4, 16となる。経過時間の比は、n=1000においてそれぞれ2.16, 6.56である。VP比を大きくすることで、通信のオーバーヘッドが隠れるものと考えられる。n>100において、経過時間はnに比例している。nに関する計算複雑度に一致している。

図2はm=16における4096PEのMP-1のmpl言語、fortranの経過時間、スーパーコンピュータS820/80の(VPU+CPU)時間を比較したものである。n=1000における時間の比は、S820/80を1.0として、MP-1のmpl, fortranに対して、それぞれ1.65, 2.67である。すなわち、エイリアス確率計算に関して、4096PEのMP-1の経過時間は、S820/80の(VPU+CPU)時間の1.65倍である。超並列計算機の効果を示している。経過時間と(VPU+CPU)時間を比較していることに注意を要する。n>100において、経過時間はnに比例しており、nに関する計算複雑度に一致している。S820/80のベクトル化率は、m=16, n=1000において99.955%である。

図3は、m=16に対して、4096PEのMP-1のmpl, fortranの経過時間、スーパーコンピュータS820/80の(VPU+CPU)時間を比較したものである。m≤12に対して、mplとfortranの差はほとんどない。m=14, 16では、PEの仮想化がおこなわれる。仮想化は、mplでは人手によっておこなわれ、fortranではコンパイラによっておこなわれる。今回の問題では、人手による仮想化の方が効率が良かった。m>10に対し、S820/80の(VPU+CPU)時間は2^mに比例している。m≤12に対し、MP-1の経過時間の増加は緩やかである。m≥14では、MP-1の経過時間の増加は2^mに近づいている。

6. まとめ

超並列計算機MP-1を用いたエイリアス確率計算の改良をおこなった。スーパーコンピュータS820/80と比較した。その結果、m=16, n=1000というパラメータにおいて、4096PEのMP-1の経過時間は、S820/80の(VPU+CPU)時間に比べて約1/1.65の性能となった。超並列計算機では、PE数を増やすことで性能が向上する。本検討により、超並列計算機の可能性を示したと言える。超並列計算機は、ソフトウェア、ハードウェアともに、まだまだ改善の余地があるように見える。アーキテクチャが若く、今後の発展が期待される。今後の課題として、他の応用評価等が挙げられる。

謝辞 MP-1の使用にあたり、株式会社理経高橋幹雄氏、宮本敏氏にお世話になりました。深く感謝します。

文献 [1] 岩崎一彦他, "超並列計算機MasParを用いたエイリアス確率の計算," 信学技報, CPS Y91-46/FTS 91-45, pp. 41-48, 1991年12月. [2] 岩崎一彦他, "スーパーコンピュータS820/80を用いたエイリアス確率の計算," 信学春大, D-116, 1992年3月.

表1 プログラムの改良項目とその効果

項目	MP-1 mpl	S820/80
出力命令の削除	1.8	1.0
べき計算の工夫	4.3	2.1
シグナル部分の改良	2.2	1.0
ショートインジックの使用	1.3	—
合計	22.1	2.1

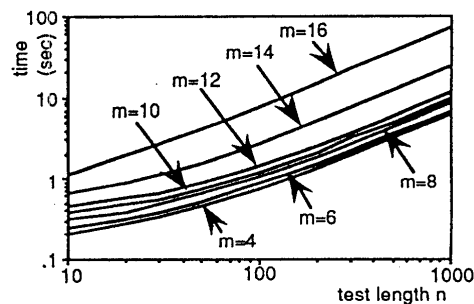


図1 MP-1のmpl言語を用いたエイリアス確率計算の経過時間。nはテスト長。mは圧縮回路の段数。

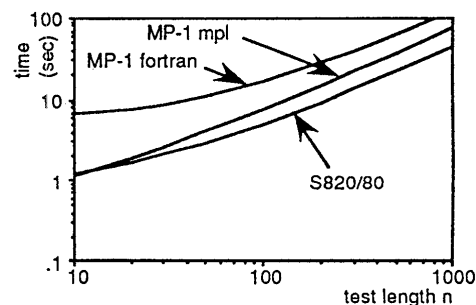


図2 圧縮回路の段数m=16におけるエイリアス確率計算時間の比較。MP-1のmpl言語、MP-1のfortranの経過時間およびS820/80の(VPU+CPU)時間。nはテスト長。

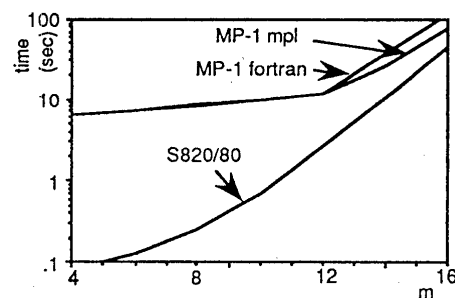


図3 テスト長n=1000におけるエイリアス確率計算時間の比較。MP-1のmpl言語、MP-1のfortranの経過時間およびS820/80の(VPU+CPU)時間。mは圧縮回路の段数。