数理解析研究所講究録
第553巻 1985年 88-99


# 科学技術計算用データ駆動計算機SIGMA－1のモンテカルロ法への適用

関口智嗣 (SEKIGUCHI, Satoshi)
島田俊夫 (SHIMADA, Toshio)
（電子技術総合研究所）

## 1．まえがき

科学技術の分野では大規模計算の高速実行を要求する度合いがその発展とともに日々に高まってきている．これらの計算は現在Cray-1,VP100,S-810などパイプライン方式の商用スーパーコンピュータで実行されることが多い．しかしこれらのスーパーコンピュータは限られた種類のベクトル演算のみを並列処理するため，必ずしも広範囲な大規模計算を高速に処理するという要求を十分に満たすことができないでいる．

その理由はパイプライン方式が算術演算やシフト演算を高速に実行するパイプを複数個用いて高速化を図っているため，これらのパイプを効率良く使用するようなベクトル計算だけしか高速に実行されないからである．

ところが現実の科学技術計算には多様なレベルの並列性があり，スーパーコンピュータが普及するにつれ商用のスーパーコンピュータでは適応できない問題の存在が浮かび上がってきている．このような状況の中でこれらの問題を高速に実行するため，マルチプロセッサ方式の計算機システムが注目を集めている．しかし従来の実験機は市販のマイクロプロセッサを多数

台結合したものが多く，その方式に適合した問題ならば並列性を十分に引き出すことも可能であるが応用範囲は限定されている．

われわれはこれらの問題を克服するため従来の方式とは全く異なるデータ駆動方式[1,2]でサブルーチンレベル，ベクトルレベル，スカラーレベルの計算すべてを並列に実行するスーパーコンピュータＳＩＧＭＡ－１[3,4,6]を開発中である．ここではその概要を説明し，従来の商用スーパーコンピュータが苦手としているモンテカルロ法への適用について述べる．

## ２．データ駆動型計算機の特徴

データ駆動型計算機とは，ある演算に必要な入力データが揃えばその演算を実行する計算機である．その特徴を実際のプログラムを用いて示す．

２次の正方行列　$X = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

が与えられたとき，その行列式

$$\det(X) = a * d - b * c$$

を求めるプログラムを考える．この計算のデータフローグラフを図１に示す．

従来のノイマン型計算機では

step　1：　e = a * d
　　2：　f = b * c
　　3：　det(X) = e − f

のようにプログラマが計算の実行順序をすべて陽に記述する必要があった．ところがユーザにとって重要なことはXが与えられたときの det（X）の値そのものであり，計算の実行順序ではない．さらに図1からも明らかなようにstep1と2の実行順序も本質的でない．

一方データ駆動型計算機では計算の実行順序をユーザが記述する必要はなく det（X）を求めるための演算を

演算　♂：　e = a＊d
　　☆：　det（X）= e − f
　　♀：　f = b＊c

として並べればその順序は問題ではない．データ駆動型計算機に入力データa，b，c，dを与えると実行可能な♂と♀をまず計算する．その結果であるe，fが揃えば☆が実行され，めでたく det（X）の値が得られる．

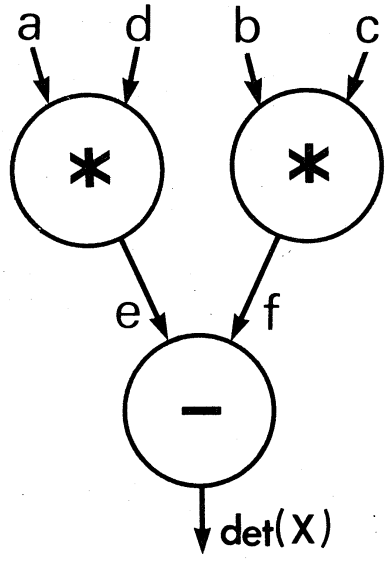

図1　Data Flow Graph of det(x)

このプログラムの並列計算の可能性を考えてみる．図1からもわかるようにeとfを求める計算は並列に実行できる．しかしノイマン型並列計算機ではどの部分を並列に実行するかを指定しなければならない．ところがデータ駆動型並列計算機では特に実行順序を示さなくても♂と♀は並列に実行される．このようにプログラムが内包する並列性を自然に抽出しながら実行するのがデータ駆動型計算機の特徴である．

この例はスカラ演算であったが，図1の'＊'や'−'という演算がもっと複雑な関数（サブルーチン）であってもよい．その場合は関数（サブルーチン）レベルの並列実行が行なわれ

る．またベクトル計算のようなループ計算であってもこれを展開することにより並列実行される．もっとも，純粋なベクトル計算ではその効率はベクトル計算機には及ばない．しかし分岐を含むようなループ計算ではベクトル計算機は効率良く処理できないがデータ駆動型計算機では良い性能を発揮することが期待できる．

われわれが現在開発中のＳＩＧＭＡ－１は以上のような特長を持つデータ駆動計算機である．

## ３．データ駆動計算機ＳＩＧＭＡ－１のソフトウェア

データフローグラフを記述すればＳＩＧＭＡ－１は自然にその並列性を抽出しながら実行する．しかし，この図そのものをプログラマが書くことは現実的でない．現在ＳＩＧＭＡ－１では高級言語としてデータフロー Ｃ言語（ＤＦＣ）を用意しており，プログラマはＤＦＣで記述すればコンパイラがデータフローグラフと等価な言語に自動変換を行なう．またＣ言語自身はオペレーティングシステムＵＮＩＸの基幹言語として良く知られているものである．

ＤＦＣはそのＣ言語と文法的には殆ど変わらない．主たる相違点は単一代入と呼ばれる規則である．この意味は同じ関数（サブルーチン）の中で同じ名前の変数に二回以上値を代入してはならないということである．たとえば同じサブルーチン内で

```
x = a + 3;
    .
    .
x = b * c ;
```

のようにｘに二度代入してはならない．しかしこれは最初のｘをｘ１，次のｘをｘ２と名前を変更するだけで本質的な問題はなく，自動変換することも可能である．

## ４．データ駆動計算機ＳＩＧＭＡ－１のハードウェア

ＳＩＧＭＡ－１はＰＥ（Processing Element）およびＳＥ（Structure Element）から構成される．これらを共に６台ずつルータネットワークでローカルに結合し１グループを形成する．更に３０グループをルータネットワークでグローバルに結合し全体を構成する（図２）．各ＰＥの性能は３ＭＦＬＯＰＳで，全体の最大性能５４０ＭＦＬＯＰＳ，また実効性能１００ＭＦＬＯＰＳを目標としている．

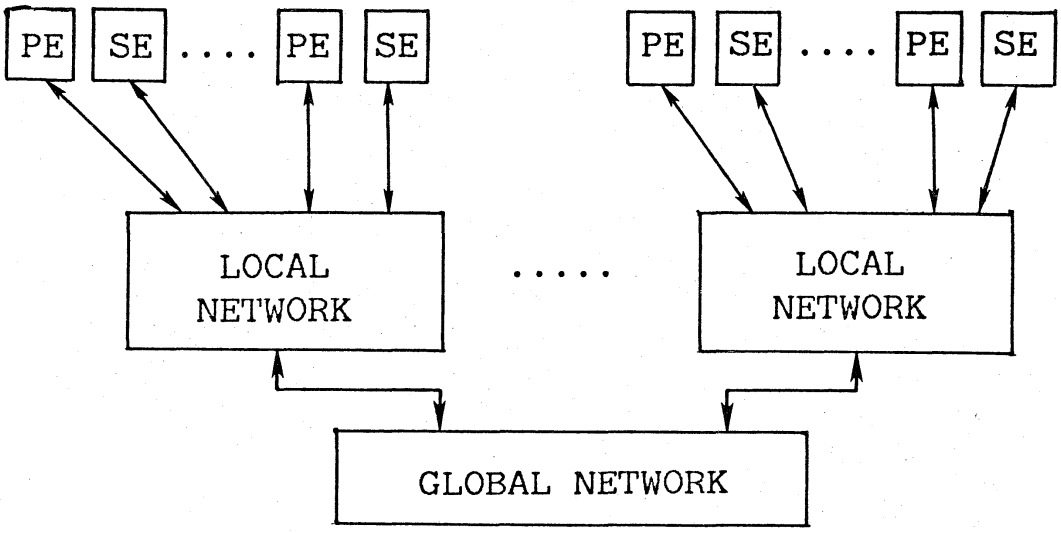


図２　　ＳＩＧＭＡ－１の構成

ＰＥは二段のパイプライン，ネットワークは二段または三段のパイプラインとなっているので，ＳＩＧＭＡ－１全体が巨大なパイプライン構成となる．ＳＩＧＭＡ－１のようなデータ駆動型計算機の構成では問題に並列性が十分あれば，パイプラインが全てデータで埋まり，ネットワークの遅延が性能低下をもたらさない．したがって実効性能の向上が期待できる．

## ５．ＫＥＮＯ概要

以上述べてきたようにＳＩＧＭＡ－１はスカラ，ベクトル，関数（サブルーチン）レベルの全てに対して並列実行を行なう．したがって，並列性のレベルが混在する問題はＳＩＧＭＡ－１の最も得意とするところである．この具体的な例題としてモンテカルロ法のＫＥＮＯコードをとりあげる．

ＫＥＮＯは原子力工学の中で臨界安全性計算を行なうための中性子輸送問題を解くコードである．すなわち中性子個々についてその発生から消滅まで（世代と呼ぶ）を約300 個の粒子について約200 世代にわたって追跡し中性子の発生／消滅の比（Ｋｅｆｆ）を求める．この問題はＫｅｆｆについての精度は要求されておらず，また幾何形状記述が容易であるという点からモンテカルロ法が用いられている．

コードの主要な部分は図３のようになっており中性子の輸送距離計算を行なうＰＡＴＨ，中性子の通過領域の判定を行なうＣＲＯＳＳ，中性子が衝突を起こした時の中性子の重みの増減処理，中性子の生成消滅処理，散乱角の計算，エネルギー群の決定，統計処理用データの収集を行なうＸＳＥＣなどのブロックがある．

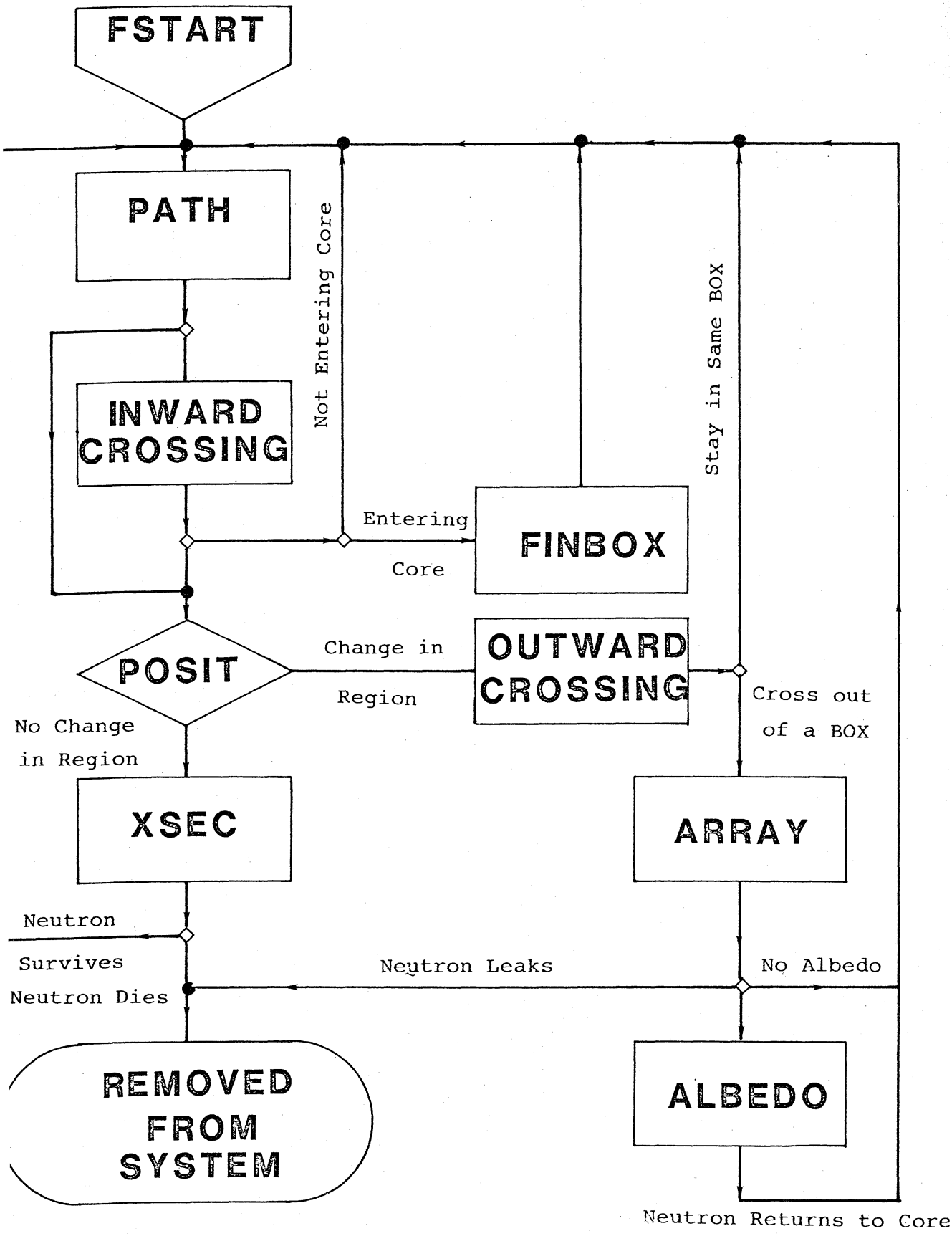


図 3 FLOW DIAGRAM of KENO

このＫＥＮＯは原子力コードで最も頻繁に用いられるものの一つであり計算機処理の高速化が望まれている．原研におけるベクトル化の試みもあるが[8]，中性子の寿命や扱いが異なるためベクトル長が減少していく，リストベクトルを扱う，例外処理が多い等の理由によりベクトル処理が難しい．

一方ＳＩＧＭＡ－１においてＫＥＮＯを並列処理するにはコードの論理的構造をプロセッサ空間上にマッピングするジョブ・アロケーションが必要である．また特にモンテカルロ法の場合には乱数の発生メカニズムや統計的処理，物理定数の参照などの問題がある．

## ６．ジョブ・アロケーション

180台ＰＥ規模のＳＩＧＭＡ－１に仕事を割り付けるにあたってその処理性能を向上させるためには各ＰＥに対して負荷分散を均一化するように，またネットワーク通過（特にグローバル・ネットワーク）に伴うオーバーヘッドを最小化するように工夫が必要である．

実行中にプロセスの割り付けを行なう非決定的なダイナミック・アロケーションは負荷分散を均一化するハードウェアのサポート[7]を得られるためユーザーはプロセスの割り付けを意識しなくてもある程度十分な並列処理が期待できる．しかしながら，ネットワーク通過によるオーバーヘッドはあらかじめ予測することができず一般にその負荷は大きめに現われる．一方実行前にプロセスの割り付けを行なうスタティック・アロケーションはユーザが負荷分散を任意に行なえるため，対象とする問題に応じて負荷分散とネットワーク負荷のバランスをとり

ながら割り付け方法の最適解を求めることが可能である．しかし分割したプロセス同志が複雑に交信し合うような構成の問題では一般に最適バランスを図ることは難しい．

ここでは６ＰＥを１グループとしたグループ毎に一定の粒子を割り当て，粒子の生成から消滅までをすべてそのグループで面倒をみるという基本的にはスタティック・アロケーションを採用する．その理由は中性子相互の振る舞いは独立であるためネットワークへの負荷が少なくて済み，またある程度の個数の粒子が集まるため負荷分散がほぼ均一に保てるからである．さらに世代の後半で idle となったグループへハードウェアサポートによるダイナミック・アロケーションを併用することでＰＥの待ち時間は少なくなる．

## ７．乱数発生機構

ここでの乱数発生は合同法を対象とする．並列計算を行なうとほぼ一斉に多量の乱数が必要となり，乱数発生を１ケ所で行なっていると他の計算に追いつかず逆にネックになってしまう．ところがすべての部分で独立に乱数列を発生すると乱数列の多次元一様性が保証されなくなる恐れがある．したがって乱数の多次元一様性が保証される範囲でできるだけ並列に計算できるようにする．ＫＥＮＯにおいて乱数は輸送距離，散乱角などを求めたり粒子の生死判定などに用いられその頻度は比較的高く全体の計算に占める乱数生成の比率は約 10%程度である．したがって乱数獲得のオーバーヘッドを考慮して全ＰＥの10から20% を乱数発生専用とすれば計算量のバランスも得られ乱数列の多次元一様性も期待できる．現在考えている方式では

$$X_{n+1} = A \cdot X_n + B \mod P$$

において生成される列を 3から6 個毎にとりあげる乱数となる．また全ＰＥによる乱数発生法やＭ系列による乱数の並列計算への適応についても検討を加えている．

## ８．統計的処理，物理定数の参照

ＫＥＮＯでは領域ｉを通過した粒子の個数を数え上げるなどの統計的処理が必要である．この場合

```
A［ｉ］＝A［ｉ］＋１；
```

のような演算となるが，配列に対しても先に述べた単一代入規則が適用されるので配列の同じ要素へ値を書き込むのも一回限りである．上式のように二回以上代入する場合は原理的に

```
B［ｉ］＝A［ｉ］＋１；
```

とし，次々と新たな配列にコピーするような措置をとらなければならない．この制約は統計的処理を行なう場合に問題となる．そこでＳＩＧＭＡ－１では範囲を限って何回でも書ける配列を用意し，ユーザの負担なしで上記のような例を救済できる．

物理定数の参照は一般に読み出し専用の配列を用意すれば十分であるためその容量に応じてグループ毎にもつか，全体で（多少の処理速度は犠牲になるが）１つの配列を分担するかを決定すればよい．

## ９．評価

以上述べてきたような方針でＳＩＧＭＡ－１においてＫＥＮＯを実行した場合の予想計算時間を算出してみる．乱数発生

を除いた全体の計算量は，M380 による測定から約４ＧFLOPと推定した．また乱数発生用ＰＥ30台を除いたＳＩＧＭＡ－１の速度は

３MFLOPS/PE＊１５０台＝４５０MFLOPS

とし，実効率ａを10-50%[5] で計算する．

（４ＧFLOP／４５０MFLOPS）＊ａ

＝　２０秒　（ａ＝50%）

＝１００秒　（ａ＝10%）

ベクトル化を行なってＶＰ100 で実行すると 380秒要する[8] ことからＳＩＧＭＡ－１でＫＥＮＯを実行した場合に商用スーパーコンピュータよりも十分高速に処理できることが期待される．

## １０．まとめ

ＫＥＮＯは商用のスーパーコンピュータにとっては非常に性質の悪い問題であるにもかかわらず，ＳＩＧＭＡ－１には十分に適用できることが明らかとなった．さらに地球物理学や高エネルギー物理学の方面などでも同様のモンテカルロ法による応用プログラムがあることがわかっている．今後はモンテカルロ法以外にもＳＩＧＭＡ－１の適用性を確認していく．

本研究を行なうにあたり日本原子力研究所浅井清氏をはじめとして計算センタの方々に貴重な助言をいただき感謝いたします．また日頃から御指導下さる電子技術総合研究所・柏木電子計算機部長，弓場計算機方式研究室長ならびに同僚諸氏に感謝いたします．

［参考文献］

[1]J.B.Dennis,"First Version of a Data Flow Procedure Language" Lecture Notes in Computer Sciense, Vol.19, Springer Verlag,1974,pp362-376.

[2]Arvind,et.al,"An Asynchronous Language and Computing Machine",TR114a Univ.of California Irvine,1978.

[3]Shimada,T.,et.al,"An Architecture of a Data Flow Machine and its Evaluation",Digest of papers,COMPCON Spring 84,1984,pp.486-490.

[4]Hiraki,K.,et.al,"A Hardware Design of a Data flow Computer for Scientific Computations" Proc 1984 of Int'l Conf. on Parallel Processing 1984,pp.126-1334.

[5] 島田，関口，平木，西田：科学技術計算用データ駆動計算機ＳＩＧＭＡ－１のシミュレーションによる性能評価，情報処理学会第29回全国大会 2C-5 .

[6] 平木，西田，関口，島田：科学技術計算用データ駆動計算機ＳＩＧＭＡ－１の予備試作，情報処理学会第30回全国大会 4C-8.

[7] 平木，関口，島田：科学技術計算用データ駆動計算機ＳＩＧＭＡ－１における負荷分散方式，情報処理学会第30回全国大会 4C-10.

[8] 浅井：スーパーコンピュータの動向と原子力分野におけるベクトル処理，第16回「炉物理夏の学校」テキスト，日本原子力学会，1984.