Vol. 34 No. 4 情報処理学会論文誌 Apr. 1993

データフロー計算機 SIGMA-1 の基本性能評価

島田俊夫[†,*] 平木 敬[†,††] 関口智嗣[†]

命令レベルデータフロー計算機 SIGMA-1 の基本性能評価を行った．プログラムは関数型言語 DFC で記述し，実行は自動並列抽出，自動負荷分散機能を用いて行った．演算性能の評価では2つのプログラムでそれぞれ 118 MFLOPS, 154 MIPS の性能が得られ，SIGMA-1 がベクトル型スーパコンピュータに比肩しうる性能を持つことを示した．通信性能は，データ分散の効果により性能が大きく影響されること，データフロー計算機の特徴であるアクセス遅延の隠蔽効果が，通信比率の高い問題では十分でないことを示した．またアーキテクチャ評価として，関数呼び出し，ループのオーバヘッドを測定した．さらにデータフロー計算機の特徴である自動並列性抽出の効果を示した．最後に待ち合わせ記憶の使用状況を解析し，記憶容量の問題について述べる．

Performance Evaluation of the Dataflow Computer SIGMA-1

Toshio Shimada,[†,*] Kei Hiraki[†,††] and Satoshi Sekiguchi[†]

Performance evaluation of the dataflow computer SIGMA-1 is presented. Programs used for the evaluation are written in the functional language DFC. Evaluation was done with automatic extraction of parallelism and automatic load distribution function. Sustained performance of floating point operations is 118 MFLOPS. This data shows that the SIGMA-1 can be comparable with existing vector supercomputers. Then the effect of data distribution between processors is measured. The results show that the feature of hiding latency of dataflow computers is not effective in case that programs have many communications. As architecture evaluations, the effect of function call, loop construct were measured and compared with a current high speed workstation. Then the effect of automatic extraction of parallelism is analyzed. Utilization of matching memory is also discussed.

1. はじめに

これまで科学技術計算はベクトル型のスーパコンピュータにより高速処理されてきた．しかしながら，ベクトル型のスーパコンピュータは純粋なベクトル計算の性能は優れているが，スカラや関数レベルの並列性を高速に処理できないという欠点がある．このため，実際の計算では，必ずしもその高いベクトル性能を生かすことができない[1]．

一方，データフロー計算機は，最も粒度の小さい命令レベルから，スカラ，ベクトル，関数のいずれのレベルでも並列性を自動的に抽出し，並列処理を行うことができる．したがって広い範囲の問題から高い並列性を引き出すことができる．とりわけ数値計算の問題は並列性が十分あることが明らかであるため，データフロー計算機により高速計算を行うことが期待できる．このようなデータフロー計算機の特長を検証するため，筆者らはデータフロー計算機 SIGMA-1 を開発した．

データフロー計算機は，古くはマンチェスタ大学で初期のデータフロー計算機を評価した例[2]があり，最近ではマサチューセッツ工科大学[3]，三洋電機[4]やシャープ[5]において開発が行われているが，その有効性を実証した例はまだない．本論文は種々の角度から SIGMA-1 を評価することにより，その有効性を明らかにしようとした．

本論文は，2章 SIGMA-1 のアーキテクチャと評価環境，3章 演算性能，4章 通信性能，5章 アーキテクチャ評価で構成されている．

† 電子技術総合研究所
Electrotechnical Laboratory
†† 東京大学理学部情報科学科
Department of Information Sciences, Faculty of Science, University of Tokyo
* 現在 名古屋大学工学部電子工学科
Presently with Department of Electrical Engineering, Electronics and Information Electronics, School of Engineering, Nagoya University

2. データフロー計算機 SIGMA-1

2.1 アーキテクチャ

SIGMA-1 システムは動的データフローモデルに基

づく命令レベルのデータフロー計算機で128台の演算処理装置 (PE), 128台の構造体処理装置 (SE), 32のローカルネットワーク, グローバルネットワーク, 16台のメンテナンスプロセッサ, 1台のホスト計算機からなる (**図 1**). 4台の PE および4台の SE はローカルネットワーク (10×10 のクロスバー) で接続され, グループを構成する. 全体は32のグループからなり, グループ間はグローバルネットワーク (2段のオメガネットワーク) により接続される.

PE は2段のショートパイプライン構造を採用し, パイプラインディレイを短くして逐次処理部の高速化を図っている. SE は, 数値計算に必要な配列の処理を行う. 配列は各語ごとに同期ビットを持ち, read と write の非同期待ち合わせが可能である. PE の性能は浮動小数点演算 2.5〜3.3 MFLOPS, その他の大部分の演算で 2.5〜5 MIPS であり, システム全体の理論的最大性能は 427 MFLOPS である. SIGMA-1 の平均命令実行時間と1パケット転送時間の比はグループ内で1, グループ間で1.7であり, 通信性能を非常に重視したアーキテクチャとなっている.

2.2 評価環境

プログラムは DFC 言語で記述した[6]. DFC は単一代入型言語でCのサブセットの機能を持っている. DFC で記述したプログラムは関数, 命令のいずれのレベルの並列性も自動的に抽出され, ユーザは並列性についての記述は行わない. ベクトル演算はスカラ命令に翻訳され自動的に並列性抽出が行われる.

負荷分散は関数レベルと命令レベルの負荷分散があり, 両方のレベルに対して静的負荷分散と動的負荷分散機能がある. 本評価では, 命令レベルは静的負荷分散, 関数レベルは, グローバルネットワークの自動負荷分散機能[7]を用いた.

プログラムはホスト計算機でコンパイルされ, SIGMA-1 にダウンロードされる. 性能測定は, SIGMA-1 プログラムの最初の命令を起動してから最後の命令を実行するまでの時間を測定し, 入出力時間は含んでいない.

3. 演算性能評価

3.1 浮動小数点演算性能

演算性能の測定は, 並列計算部分への分割が容易で計算量が多く, 通信量が少ないプログラムを用いた. プログラム1は関数 $\sin \pi x/2$ の積分を, 区間 -1 から1まで, 台形則により求める. 計算は, 区間 -1 から1までを n 区間に分割し, 分点間の面積を台形で近似してその面積を求め, 最後にそれらの総和を求める. $\sin \pi x/2$ は9次の多項式で近似した[8]. この計算は分割数だけの並列性があり, データ分散の必要がないので, 評価は自動並列抽出, 自動負荷分散機能を用いて行った. 通信は並列計算の終了時にその結果を集計するだけであるので, 通信時間は実行時間にほとんど影響しない.

PE 1台で評価を行った場合を除き, 4台の PE で構成されるグループ単位で関数を割り付けた. 従って, グループ内の PE 4台では関数内局所並列性を利用し, グループ間では関数レベルの並列性を利用することとなる. この割り付け方法は以下の評価でも同様である. なお一つのグループは最大256までの関数を

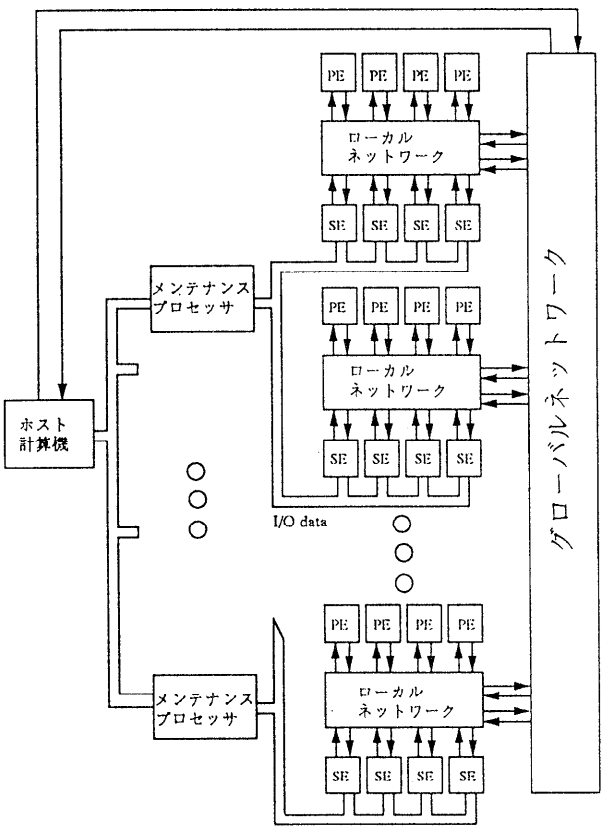


図 1 SIGMA-1 システムの構成
Fig. 1 System organization of SIGMA-1.

割り付けることができる．

プログラム1をSIGMA-1で実行した結果を**表1**に示す．比較のため，ワークステーション SUN 4/SS 2 および HP 730 のデータも示した．SIGMA-1 は SUN 4/SS 2 に比べて 22 倍，HP 730 に比べて 7.5 倍高速である．このときの SIGMA-1 の効率は 29.8% である．このプログラムにおける浮動小数点演算の比率が全体の 61.6% であることが性能低下の第一の原因である．SIGMA-1 では，実行パイプラインが1本であるので浮動小数点演算以外の演算を実行している間は浮動小数点演算を実行できない．したがって FLOPS 値の効率は浮動小数点演算の占める比率以下になる．第2の性能低下の理由は，SIGMA-1 のパイプラインにバブルが生じることである．2入力の演算の場合には，最初のデータが PE に到着したときは，対になるデータがないため次のクロックは演算ができない．プログラム1の2入力演算は全命令数の 41.6% あるので 20% 程度の効率低下の原因となる．

プログラム1を人間が静的負荷分散を行い SIGMA-1 の 128 台プロセッサで測定した結果は 170 MFLOPS であった[9]．本実験では，自動負荷分散機能を用いているが，これは人間による静的割付に比べて 1 PE の性能が 69% に低下している．性能低下の理由は，自動負荷分散では1個の関数を1グループ内の4台の PE で並列実行するので PE 間の遅延が入るが，人間が行う静的割付の場合はまとまった計算の単位を1台の PE に割り付けて，PE 間の通信を減らす工夫が優れているためである．このことは，グループ内での自動負荷分散機能をさらに効率良くすることが課題であることを示している．

3.2 演算性能（MIPS 値）

MIPS 値の測定に用いたプログラム2はモンテカルロシミュレーションのプログラムである．プログラム2は，π の値を次式に基づき，関数 $1/(1+x^2)$ の0から1までの面積として求める．

$$\int_0^1 \frac{1}{1+x^2}\mathrm{d}x=[\tan^{-1}x]_0^1=\frac{\pi}{4}$$

表 1 浮動小数点演算性能
Table 1 Performance of floating point arithmetic operations.

	SIGMA-1 (120 台)	SUN 4/SS 2	HP 730
実行時間 (msec)	1.18	24.3	8.24
MFLOPS 値	118	5.3	15.7

求め方は，1辺が1の正方形の中で一様乱数 (x, y) を生成し，そのときの y の値が $1/(1+x^2)$ の上になるものと下になるものの比率を調べる．正方形の面積は1であるので，その比が $\int_0^1 \frac{1}{1+x^2}\mathrm{d}x$ の面積となり，π の値が求まる．乱数発生は，直前に発生した乱数をその次の乱数発生の種に用いる再帰計算である．ここでは，乱数発生の関数を複数個用い，それぞれに異なる初期値を与え，乱数発生関数を並列に計算した[10]．プログラム2はプログラム1に比べて，通信を行う命令を 6% 程度含んでいるという特徴がある．また関数レベルの並列性はあるが，ベクトル計算機での高速化が難しい問題である．

プログラム2を SIGMA-1 で実行した結果を**表2**に示す．ここでもワークステーション SUN 4/SS 2 および HP 730 との比較を行った．SIGMA-1 は SUN 4/SS 2 の 18.4 倍，HP 730 の 3.9 倍高速である．

プログラム2のグローバル通信の影響を見るため台数効果を**図2**に示した．プログラム2では4台で 2.5 倍，8台で 4.8 倍，16台で 9.5 倍，32台で 18 倍，64台で 37 倍，120 台で 70 倍の性能であり，十分な台数効果が得られた．プログラム2はグローバル通信命令を約 6% 含んでおり，グローバル通信命令は通信に閉塞が全くない場合で平均命令実行時間の 5.3 倍の実行時間を要するにもかかわらず，通信遅延がオーバヘッドになるには至っていない．

表 2 演算性能（MIPS 値）
Table 2 Performance of operations (MIPS).

	SIGMA-1 (108 台)	SUN 4/SS 2	HP 730
実行時間 (msec)	1.66	30.5	6.57
MIPS 値	154	8.4	39

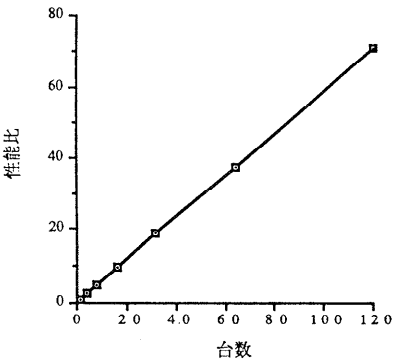


図 2 モンテカルロ問題を実行したときの台数効果
Fig. 2 Scalarbility of performance for Monte Carlo problem.

4. 通信性能

SIGMA-1 の通信性能は，ローカルネットワークの遅延が1クロック，グローバルネットワークの遅延が2クロックである．ローカルネットワークとグローバルネットワークの最大転送性能はそれぞれ，600 MB/sec, 1.9 GB/sec である．なお SIGMA-1 の1クロックは 100 ns である．プロセッサ間の通信は行く先プロセッサのバッファに到着するまでにローカル通信で2クロック，グローバル通信で5クロックの遅延がある．パケットは上位49ビットと下位40ビットを分けて転送するので，パケットが完全に到着するにはもう1クロックかかる．また SE のデータ構造のアクセス処理時間は3クロックである．

通信性能を評価するため，プログラム3として SOR 法による偏微分方程式の数値解法の中心部分を取りだした．格子状に配置された節点の値を，上方および左方からのすでに更新されている値，当該節点，右方および下方からの更新前の値を使用して求めるものでウェーブフロントアルゴリズムとして知られているものである．並列性は行列の対角線方向に存在し，隣接する節点から送られる更新された値により評価が開始される．本プログラムは正方格子でない格子形状を持つ場合，ベクトル計算機によるベクトル化が困難であり，並列計算機による実行では，行列の要素レベルにおける細粒度同期が必要であり，通信／同期オーバヘッドがプログラム2より大きな問題となる．

プログラミングは，行列を1列ごとに切りだし，処理装置を割り当て，格子間の通信は Bデータ構造[6]を使用し，2個のデータ構造を交互に使用して演算を行った．この場合，列の数と同数の関数を並列実行する．

プログラム3の評価では，並列処理部分，データ構造の配置ともハードウェアの持つ自動負荷分散機構を使用した．SIGMA-1 の自動負荷分散機構では，関数およびデータ構造は，割り付け実行時に最も割り当て数の小さいグループに割り付けられる．また関数と，関数内部で使用するデータ構造は互いに独立に割り付けられる．従って，ほとんどすべてのデータ構造アクセスはグローバルネットワークを経由し，グローバル通信が非常に多い問題である．グローバルネットワークは各グループと2本のチャネルで結合し，さらに2段のオメガ網であるため，入出力および内部ノードでの閉塞が発生しやすい．

SIGMA-1 システムで利用するプロセッサ数が少ない場合には，配置に自由度がある．ネットワークの閉塞が性能におよぼす影響を見るため，グローバルネットワークで閉塞が多い配置，閉塞が少ない配置をヒューリスティクスで求め，性能を測定した結果を**図3**に示す．なお，プロセッサ台数 32 台以上では閉塞が少ない配置を得ることができないため測定を行わなかった．

図3では 16 台のプロセッサを用いた場合，最悪の配置で約 44% の性能低下が見られる．従って，グローバルネットワークにおける飽和が性能低下の大きな原因であることが推定される．データフロー計算機の大きな特徴の一つは，グローバルアクセスの間に他の実行可能な命令を実行することにより，グローバルアクセスの遅延を隠蔽できることであるが，プログラム3のように通信の比率が大きくなると，この機能だけで性能低下を防ぐことはできない．通信の比率が高いプログラムではデータフロー計算機でも，グローバルアクセス回数を減少させるようなデータの分散配置が重要であることが分かる．

5. アーキテクチャ評価

5.1 関数呼び出しの評価

DFC は関数を並列処理部分の単位とし，グループ間の分散を行うため，並列処理部分を増やそうとすると，通常のCプログラムと比較して多数の関数呼び出しを行うことになる．SIGMA-1 における関数呼び出し／復帰に関するオーバヘッド，特にグループ間で引数や，返値を通信する際の関数に関するオーバヘッドは性能低下の原因と考えられる．

測定はカウンタを再帰的に構成する簡単な再帰関数を用いて関数1回あたりの所要時間を比較した．プロ

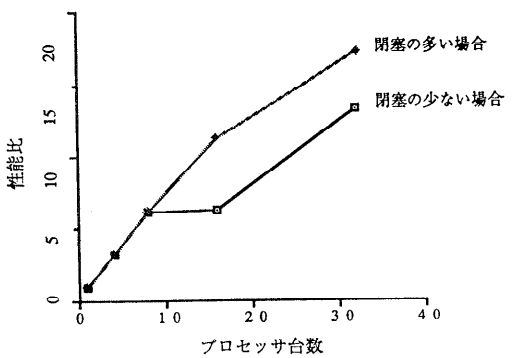


図 3 データ分散の効果
Fig. 3 Effect of data distribution.

グラムは以下のとおりである.

```
count(n)
  int n;
  {int m;
    if(n<2) m=1;
    else m=1+count(n-1);
    return(m);
  }
```

再帰呼び出しを1000回まで行って測定した結果，関数1回あたりの所要時間はSIGMA-1の1PEで10.4 μsec，HP 730は0.275 μsecである．またSIGMA-1の1回の関数呼び出し／復帰は約7 μsecかかる．両者の比は約38倍であるが，比較のためには両者のクロックがそれぞれ10 MHzと66 MHzで6.6倍の差があり，HP 730は1クロックに1～2命令の実行が可能であり，SIGMA-1は平均的には3クロックに1命令実行であることを考慮する必要がある．そこで関数本体部と関数呼び出し部の実行時間の相対比を比較した（**図4**）．その結果，HP 730はSIGMA-1の1PEの関数呼び出しより約28%小さく，関数呼び出しはSIGMA-1の性能低下の要因となり得ることを示している．この差は関数引数をレジスタ／スタックで渡す方法とパケットとして渡す方法の差と考えられる．

しかし1グループで実行した場合は，カウンタプログラムは逐次的なプログラムであるにもかかわらず，命令レベルの並列処理効果により若干の性能向上がみられ，HP 730との差は小さくなる．

5.2 ループ

SIGMA-1はループ制御のために通常3命令を使用する．これはノイマン計算機に比べて多く，ループのオーバヘッドとなる．このオーバヘッドを測定した．

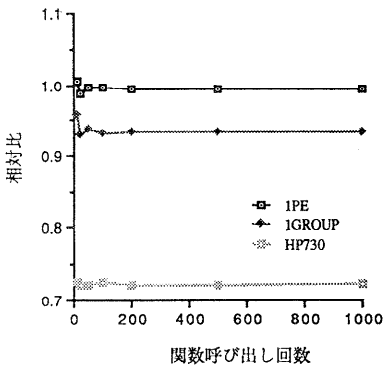


図4 関数呼び出しの比較
Fig. 4 Comparison of function ball.

測定は通常の内積プログラムと，そのループボディを10回展開し繰り返し回数を1/10にしたプログラムの実行時間を比較して行った．また同じプログラムをHP 730で実行した結果も示す（**表3**）．

SIGMA-1のオーバヘッドはHP 730に比べて約11倍大きい．この差は，両者のクロックに6.6倍の差があること，HP 730は1クロック1～2命令実行が可能であること，レジスタの使用，実行命令数の差などが主たる原因と考えられる．また同じプログラムを1グループ（4PE）で実行した場合は，ループボディを展開しないプログラムの方が実行時間が短くなっており，ループオーバヘッドが並列処理により隠されたと考えられる．その理由は，並列処理をすることによりどの時点でも実行可能な命令数が増し，SIGMA-1の実行パイプラインのバブルが減少するからである．このことは，ループオーバヘッドはSIGMA-1では大きな問題でなく，むしろ命令レベルの負荷分散の方が性能に大きな影響を与えることを示している．

5.3 自動並列性抽出

データフロー計算機は単一代入型言語を用いるため複雑な構造のプログラムでも並列性を自動的に抽出できることが大きな特徴となっている．この特徴を評価するためループと再帰呼び出しを持ったプログラムを用いた．その理由は，ループと再帰呼び出しが大きな並列性を生み出す基本構造であり，これらの並列性を効率良く処理できることが並列処理において重要だからである．

実用的問題の多くは，ループがネストしており，その中には条件文やベクトル計算，関数呼び出しがあり，しかもそれらがデータ依存関係を持っている．ベクトル計算機はこのような複雑な構造を効率良く処理することができない．

一方SIGMA-1はどのように複雑な問題であっても全体を一つの大きなデータフローグラフと見なして処理を進めるので，問題に並列性がありさえすれば並列処理は容易である．しかも自動負荷分散やユーザ指定の関数レベル負荷分散が簡単に行える．これは関数がそれ自体閉じた局所処理であり，副作用を利用していないところからくるデータフロー方式に特有の利点

表3 ループ制御の実行時間
Table 3 Execution time of Loop construct.

	SIGMA-1 (1 PE)	HP 730
ループ制御実行時間 (μsec)	1.0	0.09

である．本節では，ループの中に再帰関数がある複雑な構造を持つ問題の並列性抽出を取り上げ，SIGMA-1 の並列性抽出能力を示す．

問題はコインの種類による場合分けで，ここではコイン問題と呼ぶ．この問題は，条件によって状況を分類し計算を進める．このような場合ループで問題を表現すると条件文が複雑にネストし，プログラムの記述や解釈が容易でなくなる．このような問題は再帰型のプログラムで記述すれば，プログラムの記述が容易で見通しが良くなる．このようなプログラムはベクトル計算機では全く並列処理できないが，並列性を十分に有する問題である．具体的には分割統治型の再帰関数により問題を記述し，それらの関数を自動負荷分散機能を用いて負荷分散し，実行する．

コイン問題は，ある金額のお金を，1円，5円，10円，100円，500円の各種のコインで持つときに，どのような組み合わせがあるかを数え上げる問題である．問題の解き方は，まず一番大きいコインが最大 n 個持てるとすると，そのコインを0個から n 個まで持った場合に分け，次にその各場合について，2番目に大きいコインの持ち方が何通りあるかを，一番大きいコインの場合と同様に場合分けする．この方法を繰り返し，一番小さいコインに達したところで，すべての場合の数を数え上げれば答えが得られる．DFC で記述した関数 count の主要部を下に示す．

```
if(coin <= 0) n = fit(amount, val);
else
  for (i=0, n=0; i<= lim; i=i+1, n=newn)
      newn=n+count(amount-(i*val),
        coin-1, coinvalues);
```

関数 count が再帰的に呼ばれ，上述の分割統治部分を表現している．関数 count は第1引数として金額，第2引数として一番金額の大きいコイン，第3引数としてコインの種類を表す配列を取り，返値は n の値である．関数 fit は，第1引数で指定される金額を第2引数で指定されるコインで，余りなしに持てる場合は1，余りがあるときは0を答えとして返す関数で，このプログラムでは終了判定に使われている．

count 内のループは，最大金額のコインを持てる場合の数だけ繰り返しを行い，関数 count を再帰的に呼び出す．これを 500 円，100 円，…1円のコインまで繰り返すので，関数 count の中のループがネストする回数をそれぞれ $n_1, n_2, \cdots, n_k$ とすると，$n_1 \times n_2 \times \cdots \times n_k$ 回関数 count を呼び出し，n 分木を作る．n 分木のリーフにあたる部分が並列処理されるが，実際には関数展開の遅れなどから展開した木の左下から右上に向かって並列処理が行われると考えられる．

たとえば 100 円を 50 円以下の種類の硬貨で持つ場合の数は何通りあるかという問題では，関数展開の n 分木は**図5**のように表現され，実際の実行は1円硬貨を調べる左下から右上に向かって処理が進むことになる．

このプログラムの実行で難しい問題は，実行状況のトリーの各レベルで場合分けの数が実行時に動的に決まるために，静的負荷分散が困難なことである．しかしながら SIGMA-1 ではネットワークが動的負荷分散機能を持つため，動的に生成される関数を効率良く並列処理することができる．

コインプログラムを実行するときの SIGMA-1 の状況をシミュレーションで解析した．問題は 200 円を 50 円以下のコインで持つときの場合を数えている．**図6**に関数レベルの並列性として，実行中に存在する関数の数を示す．存在する関数の数は，最初かなり急激に立ち上がり，ピーク時は 482 の関数が存在している．その後関数の数は徐々に減少する．関数の数が128以上ある期間はサイクル400から2600までで全実行時間の 65% を占めている．しかしこの問題の関数は再帰呼び出しをしているため，図5のトリーの上位レベルの関数は下位レベルの関数を呼び出すと待ちの状態に入る．そのため，実際の並列性は関数の数ほどにはない．このプログラムでは関数呼び出しは 2115 回行われ，図6のグラフに示すように関数呼び出しの最大

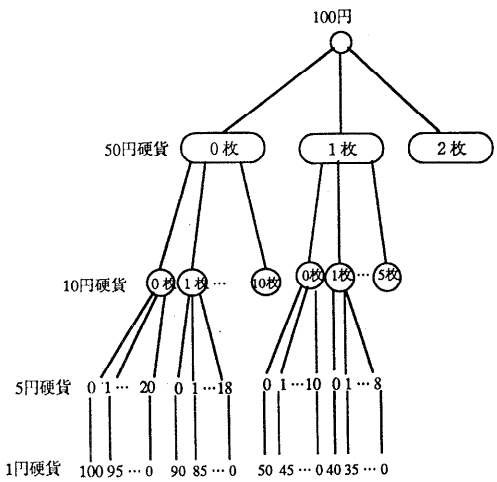


図 5 コイン問題の実行状況
Fig. 5 Execution profile of coin problem.

値は約500であり，SIGMA-1 の128台のプロセッサに十分な並列性抽出を行っている．

次に命令レベルでの並列処理の様子を調べた．**図7**に平均実行命令数とトークン数を示す．実行命令数の推移は，図6の関数のグラフより早く立ち上がる．その後実行命令数は45から25へ落ちる．実行命令数が減るのは，この時期に関数が非常に増えているので関数の呼び出しと他のグループへの実行依頼のための通信時間のオーバヘッドと推定される．その後再び47まで上昇し，最後に急激に減少する．この問題でのプロセッサの平均使用率は約22%であった．この使用率は，さらに大規模な問題を実行すれば並列性が増し，通信のための待ちによるアイドル時間に実行できる命令が増すため改善されると考えられる．ここで強調したいことは問題の構造を素直にプログラムに記述し，そこから並列性を引き出した点にある．ベクトル計算機の場合，まず計算機に適したデータ構造があり，それに合わせてアルゴリズムを考えなければならない．

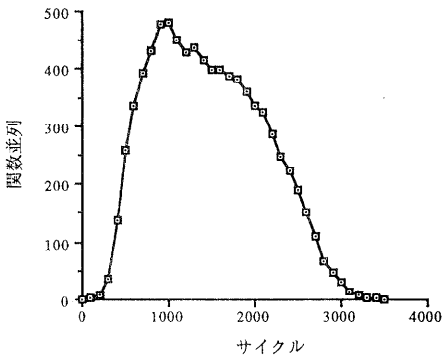


図 6 関数並列性
Fig. 6 Concurrency of function.

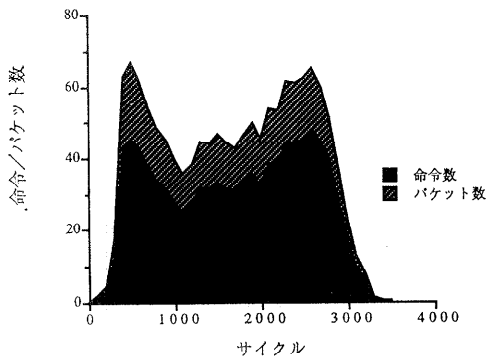


図 7 平均実行命令数と平均パケット生成数
Fig. 7 Average number of instruction execution and packet creation.

このようなプログラミングのオーバヘッドはプログラマに大きな負担を強いることになる．データフロー方式は問題をそのまま計算機上にマッピングできる点が優れているといえる．

5.4 待ち合わせ処理性能

SIGMA-1 は命令レベルデータフロー計算機であるため，同期は待ち合わせ記憶を用いて行う．2入力命令の同期は，1番目の入力データ到着時に3クロック，2番目の入力データ到着時も3クロックであるが，2番目の入力データ到着時は命令実行と動作がオーバラップするので，実質的に2クロックである．

SIGMA-1 の待ち合わせ記憶部は連鎖ハッシュ方式を採用している．この方式はメモリの占有率が高くなると，新しいエントリーを登録するとき衝突の頻度が高くなり，待ち合わせ記憶部の処理時間が増大して実行効率が低下する．したがって待ち合わせ記憶部の記憶容量は，性能を決定する際の重要なパラメータである．このパラメータを決定するためには，データフロープログラムにおけるパケットの振る舞いを調べる必要がある．

コイン問題のプログラムにおける図7の平均実行命令数と平均パケット生成数のグラフが相似であることから，平均的にはトークンは生成されると比較的短期間の後に消費されることがわかる．同じプログラムの待ち合わせ記憶部のトークンの滞留数のシミュレーション結果を**図8**に示す．

このグラフは関数の並列処理のグラフとほとんど同じである．このことは，結局，関数の生成に応じてその引数と局所変数が待ち合わせ記憶部にストアされ，それらが終了するとパケットも消費されることを意味している．分割統治型の問題は先にも述べたように，そ

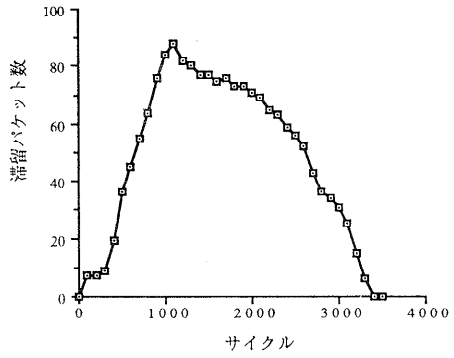


図 8 待ち合わせ記憶部のパケット滞留数
Fig. 8 Number of packets in the matching memory.

のネスト数を k とすると $n_1 \times n_2 \times \cdots \times n_k$ の関数を実行するので，大規模な問題を実行すれば大容量の待ち合わせ記憶部でも容量が不足することになる．SIGMA-1 の待ち合わせ記憶は 64 K 個のエントリーを記憶できるので，中規模の計算では不足することはないが，非常に大規模な問題を処理するためには，並列性制御の技法を用いて関数の生成を制御することが必要であることが分かる．

6. おわりに

データフロー計算機 SIGMA-1 の基本性能評価として，演算性能，通信性能，アーキテクチャを評価した．SIGMA-1 は，第一世代の命令レベルデータフロー計算機であるが，その演算性能 118 MFLOPS は現在のスーパコンピュータに十分比肩しうるものであり，データフロー計算機は効率が悪いという定説を打ち破った意義は大きい．並列性が十分にある場合は，命令パイプラインのバブルがそれほど発生せず，かなり効率良く動作することが確認できた．しかし2入力命令の待ち合わせ時に起こるパイプラインブレークは避けられない．この点を改良するためには，並列処理の粒度を大きくし，待ち合わせの回数を減らすことが考えられる．

通信性能に関しては，データフロー計算機がグローバルアクセスの遅延を隠蔽できるかどうかを調べた．その結果，通信の比率が高い問題ではその隠蔽機能は十分でなく，データ分散による通信量の減少がより重要であることを明らかにした．

アーキテクチャ評価では，関数単位での並列性抽出の結果，関数呼び出し回数が増すと性能低下の原因になることがわかった．一方ループのオーバヘッドは，それほど大きなものでないこと，1グループで並列処理すれば，ループボディの持つ並列性のため，実行パイプラインのバブルが減り，ループのオーバヘッドは非常に小さいものになることを示した．

さらに SIGMA-1 の特徴である並列性の自動抽出機能と自動負荷分散機能が十分に効果を表していることを明らかにした．大量の並列性はループと再帰構造により生み出されるので，これらの構造が複雑に入り交じっても並列処理を効率良く行うことが実用的な並列計算機には要求されるが，データフロー計算機はこの要求を満たす性質を持っている．

待ち合わせ処理性能では，大部分のパケットの滞留時間は短いため，待ち合わせ記憶は多くの場合大容量を必要としない．実際 SIGMA-1 の待ち合わせ記憶は大容量であるため中規模の問題に対しては占有率による効率の悪化は経験していない．しかし再帰的に並列性を生み出す問題では，生成する関数の数に応じてパケットの滞留数が増すので，この種の問題では並列性制御の技術を導入して，パケットの発生を制御することの必要性が確認された．

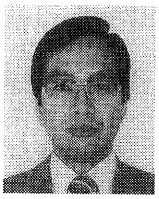
データフロー計算機の研究は，並列処理において通信と演算を融合した実行パイプラインや関数型データ構造が非常に重要であることを示してきた．今後も新しい並列処理アーキテクチャに必要な技術開発の手掛かりを得る目的で，SIGMA-1 をさらに種々の観点から評価していきたい．

謝辞 本研究を遂行するにあたり御指導，御検討いただいた電子技術総合研究所情報アーキテクチャ部長弓場敏嗣氏，熱心に御討論いただいた計算機方式研究室の同僚諸氏に感謝いたします．

参考文献

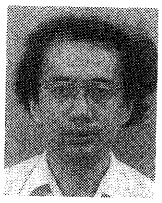
1) Lubeck, O. M.: Supercomputer Performance: The Theory, Practice, and Results, *Advances in Computers*, Yobits, M. C. ed., Academic Press, Vol. 27, pp. 310-362 (1988).
2) Gurd, J., Kirkham, C. C. and Watson, I.: The Manchester Prototype Dataflow Computer, *Comm. ACM*, Vol. 21, No. 1, pp. 34-52 (1985).
3) Papadopoulos, G. M. and Culler, D. E.: Monsson, *Proc. 17th Int. Symp. on Computer Architechure*, pp. 82-91 (1990).
4) 岡本，川口，三浦，清水：データ駆動計算機 EDDEN の基本性能評価，並列処理シンポジウム JSPP '92, pp. 337-344 (1992).
5) 内藤ほか：データ駆動型マイクロプロセッサの性能評価，並列処理シンポジウム JSPP '89, pp. 329-334 (1989).
6) 島田，関口，平木：データフロー言語 DFC の設計と実現，電子通信学会論文誌，Vol. J 71-D, No. 3, pp. 501-508 (1988).
7) 平木，関口，島田：並列計算機におけるネットワークを用いた動的負荷分散機構，電子通信学会論文誌，Vol. J 69-D, No. 2, pp. 180-189 (1986).
8) 山内，森口，一松：電子計算機のための数値計算法 I，培風館 (1960).
9) 関口，平木，島田：科学技術計算用データ駆動計算機 SIGMA-1 全体システムの評価（その2），第 36 回情報処理学会全国大会論文集，6 B-2 (1988).
10) 宮武，中山：モンテカルロ法，日刊工業新聞社 (1962).

（平成4年 9 月 28 日受付）
（平成4年12月 10 日採録）



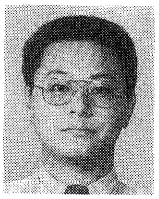
島田　俊夫（正会員）

昭和20年生．昭和43年東京大学工学部計数工学科卒業．昭和45年東京大学大学院修士課程修了．同年電気試験所(現電子技術総合研究所)入所．同所計算機方式研究室室長を経て，平成5年から名古屋大学工学部電子工学科教授．コンピュータグラフィックス，人工知能向き言語，LISPマシン，データフロー計算機の研究に従事．高度な並列処理方式に興味がある．工学博士．市村賞受賞．電子情報通信学会，日本ソフトウェア科学会，IEEE会員．



平木　　敬（正会員）

昭和51年東京大学理学部物理学科卒業．昭和57年同大学院理学系研究科博士課程修了．同年電子技術総合研究所入所．理学博士．計算機アーキテクチャ全般，特にリスト処理計算機，データフローマシン，スケジューリングなどの研究に従事．元岡賞．市村賞各受賞．情報アーキテクチャ部計算機方式研究室主任研究官を経て，平成3年から東京大学理学部情報科学科助教授．昭和63年から平成2年までIBMワトソン研究センター招聘研究員．



関口　智嗣（正会員）

1959年生．1982年東京大学理学部情報科学科卒業．1984年筑波大学大学院修士課程理工学研究科修了．同年電子技術総合研究所入所．現在，情報アーキテクチャ部計算機方式研究室主任研究官．計算機アーキテクチャと数値解析の研究に従事．特に科学技術計算用並列アルゴリズムに興味を持つ．市村賞受賞．現在，データ駆動計算機とスーパコンピュータ評価技術の研究を行っている．日本応用数理学会，日本ソフトウェア科学会各会員．