分散形データフロー計算機のシミュレーションによる設計と評価†

大山敬三†† グェン・ニュット††
斉藤忠夫†† 猪瀬博††

データフロー計算機は，従来のフォンノイマン型計算機に代わり，高度の並列性が要求される情報処理の面で，重要な役割を果たすものと期待されている．その実現に当たっては，命令を命令メモリから処理装置へ転送する裁定回路網がボトルネックとなる．筆者らは，この点に着目して，裁定回路網のボトルネックを解消する分散形データフロー計算機の新しい構成を以前から提案している．この構成に基づき，データを処理装置から命令メモリへ転送する分配回路網に共通バスを用いた縮小モデルの試作システムの設計と評価を行った．本論文では，提案システムの構成に基づいて作成したシミュレータのモデルについて述べる．また，このシミュレータを用いて，目標の性能を得るために要求される各部の速度，性能等を求め，設計のためのパラメータを得ている．この結果，分配回路網の共通バスに要求される速度や，処理装置内部の各部の処理に許容される時間等が明らかとなり，提案システムの実現性を確認できた．

1. まえがき

最近の情報処理技術はマイクロエレクトロニクス技術によって大量に安価に製造できるようになった複雑な論理機械を中心として発展してきているが，いわゆるフォンノイマン型の構造で大型計算機を構成しようとするとき，現在利用できるマイクロエレクトロニクスの量産性の特質がこれに生かしにくくなっている．

このため，小型のプロセッサを多数有機的に結合して，大型機に匹敵する能力を実現する技術開発が求められている．この実現のための有力な方法として，データの流れのみにより処理の実行を制御する，データ駆動原則に基づくデータフロー計算機 (DFC) が注目されている．

DFC の構成としても種々のものが提案されているが，その機能をデータの流れに基づき整理すると**図1**のような構成として捉えることができる．命令セルはデータフロープログラム (DFP)，すなわち，処理単位であるアクタに関する制御情報を管理する．裁定回路網 (arbitration network) は，アクタが実行可能になったときに，その実行に適した処理装置を選択して必要な情報を与える．分配回路網 (distribution network) は，処理装置におけるアクタの実行結果を受けて，そのデータの目的のアクタが格納されている命令セルに転送する．

DFC においては，データ駆動原則の自然な実現形態としておもに集中形構成が研究されている．しかし，集中形構成では，アクタを実行する処理装置を，実行時に動的に割り付ける機能が裁定回路網に必要であり，これが処理性能上のボトルネックとなる[1]．

この解決方法として，筆者らは**図2**に示すような分散形の構成を提案し[2),3)]，高性能なシステムの実現を目指して詳細な構成方法を明らかにするとともに，分配回路網に共通バスを用いた縮小モデルの試作を通して検討を行ってきた[1)]．

提案システムは，数値計算等のように，比較的固定的な応用に対して，マイクロプロセッサを多数用い，低コストで大型機なみの処理性能を実現することを目標としている．

本論文では，提案システムをモデル化し，計算機シミュレーションを行うことにより，その実現に当たって必要とされる設計の諸パラメータを導出し，また全体の動作解析を行っている．

2. シミュレーションモデル

本章では，提案システムの構成について述べ，次にシミュレーションモデルとして，データフロー計算機のモデル (DFC モデル) とデータフロープログラムのモデル (DFP モデル) とについて述べる．シミュレーションは DFC モデルが DFP モデルを実行する形式で進められる．

2.1 提案システムの構成

筆者らが提案している分散形 DFC の全体構成を**図3**に示す．このシステムでは，大型機並みの処理能力

† Design and Estimation of a Distributed Data Flow Computer by Means of Simulation by KEIZO OYAMA, NHUT NGUYEN, TADAO SAITO and HIROSHI INOSE (Department of Electrical Engineering, Faculty of Engineering, University of Tokyo).

†† 東京大学工学部電気工学科

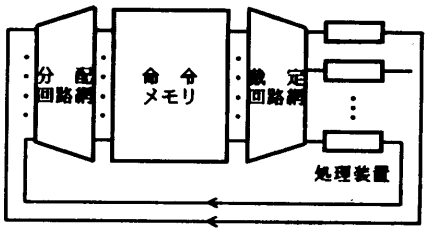

図 1 集中形データフロー計算機の構成
Fig. 1 Structure of a concentrated data flow computer.

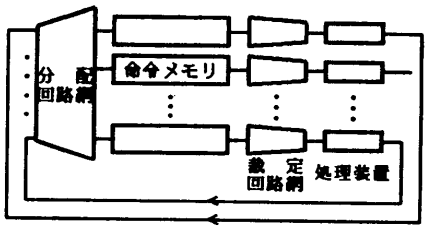

図 2 分散形データフロー計算機の構成
Fig. 2 Structure of a distributed data flow computer.

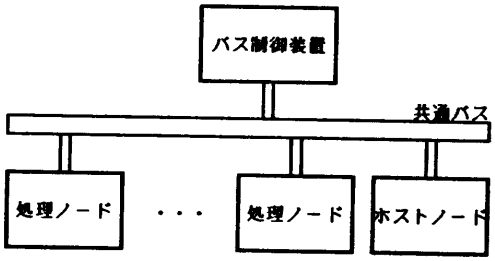

図 3 提案システムの全体構成
Fig. 3 Overall configuration of proposed system.

を実現するために，高性能マイクロプロセッサ程度の演算装置をもつ処理ノードを256台結合することを想定している．命令セルは各処理ノードに分散して配置され，アクタはその命令セルが格納されている処理ノードにおいて実行される．

共通バスは同期式で，信号線は，データ128本，アドレス8本と制御線数本を張る．

各ノードが共通バスにアクセスする際の競合制御はバス制御装置により行う．バス制御装置は競合制御回路と同期化回路により成る．競合制御回路は図4に示すように，4入力の先着順アービタを4段の階層構成として実現する．

処理ノードの構成を図5に示す．更新装置はマイクロプロセッサと付属の回路により構成される．受信装置は共通バスから受信したメッセージのヘッダを解析し，データメモリ管理装置 (DMMU) からデータメモリ (DM) の空きブロック番号を受け取り，データメモリ転送装置 (DMAC) に与えることにより，メッセージのデータを DM へ転送させる．同時に，到着メッセージの目的とするアクタの命令セルを更新する．命令セルはプログラムメモリ (PM) に格納されている．このアクタが実行可能となると，アクタの演算コードとオペランドデータの所在情報を命令キュー (IQ) に，出力データをメッセージとするためのヘッダをヘッダキュー (HQ) に入力する．

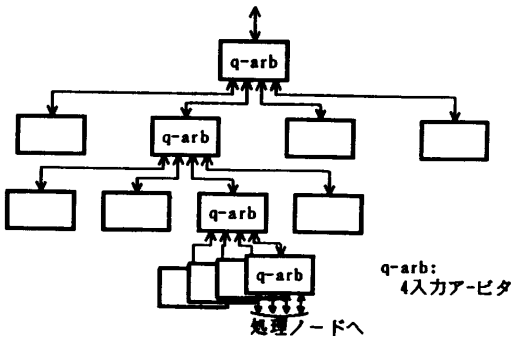

図 4 競合制御回路の構成
Fig. 4 Configuration of competition control circuit.

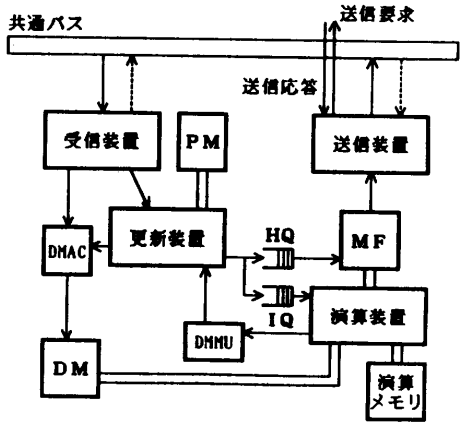

図 5 処理ノードの構成
Fig. 5 Configuration of processing node.

演算装置もマイクロプロセッサにより構成される．IQ から実行可能となったアクタの情報を取り出し，DM から必要なオペランドデータを読み出しながら演算を実行する．結果は，順次メッセージフォーマッタ (MF) に書き込まれ，アクタの実行が終了すると終了信号を与えて次のアクタの実行に移る．

MF は結果のデータに HQ から取り出したヘッダを付加し，メッセージ形式として送信装置へ転送する．

送信装置と受信装置は，共通バスの送受信手順に従いメッセージの転送を行う．

DMMU は，DM を最大メッセージ長の大きさのブロックに分割して管理し，更新装置からの要求に従い空きブロック番号を与え，演算装置で使用済みとなったブロックを回収する．

提案システムは分散形であり，アクタを記憶する命令メモリとアクタを実行する演算装置が同一処理ノード内になければならず，DFP の実行に先立ち，各アクタを処理ノードに割り付けておく必要がある．このように，分散形 DFC では割付けが静的となるため，割付けアルゴリズムがプログラムの実行効率に大きく影響する．しかし，応用として比較的固定的なプログラムを想定しており，分配回路網には局所性のない共通バスを使用するため，単純な割付けアルゴリズムで十分に高い効率が得られる．

2.2 データフロー計算機のシミュレーションモデル

シミュレーションのための DFC モデルの全体的構成は，図3に示した提案システムと同様である．バス制御装置のモデルを**図6**に示す．バス制御装置は階層構成の競合制御回路をそのままモデル化している．各段において，一つの入力源からは一つの送信要求しか受け付けない．

処理ノードのモデルを**図7**に示す．2.1 節で述べた構成と異なるのは，アクタの具体的な処理内容を考えないため，実際のデータに関する部分を省略した点である．したがって，データの DM への転送は，このモデルでは DMAC の占有という形をとり，DM と DMMU を省略している．また，MF は実際には非常に高速に処理を行うため，演算装置の処理時間に比べ無駄時間が無視しうるので，DFC モデルでは省略している．処理ノードのモデルには受信，命令，送信の3種類のキューがあり，容量は実際は有限であるが，ここでは無限長として扱う．

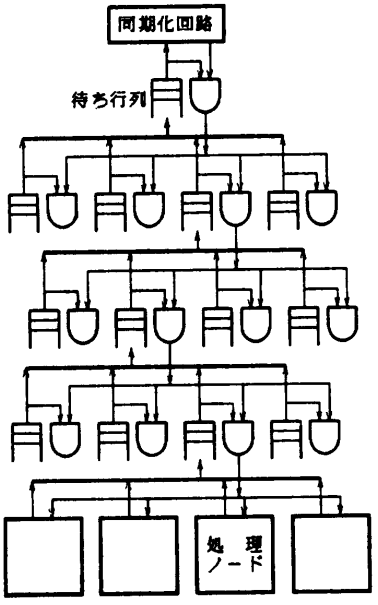

図 6 バス制御装置のシミュレーションモデル
Fig. 6 Simulation model of bus controller.

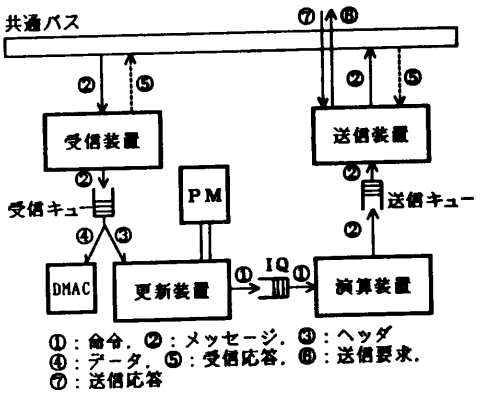

図 7 処理ノードのシミュレーションモデル
Fig. 7 Simulation model of processing node.

2.3 データフロープログラムモデル

DFC は応用により効率が変化する．とくに，プログラムのもつ並列性とアクタ間のデータ依存性が大きく影響を与える．したがって，DFP を一般化し，抽象化してシステムを評価することはむずかしい．このため，本シミュレーションでは，DFC モデルに基づくシミュレータが，与えられた DFP モデルを実行する形式でシミュレーションを進める．

DFP モデルは，データの流れを示すデータフローグラフ (DFG)，各アクタの実行時間，メッセージ長，割付け情報から成る．

DFG において DFC の実行効率に与える影響が大きいのは，アクタ間のデータの流れを示すリンクの，1アクタへの入出力数である．

この入出力リンク数は通常の数値計算等では5程度までと考えられるので，今回のシミュレーションではこれを 2, 3, 5 とし，それぞれの代表的な例として FFT モデル，1次元微分方程式モデル (DEQ 1 モデル)，2次元微分方程式モデル (DEQ 2 モデル) の3種類を DFP モデルとして採用した．**図 8～10** にそれぞれの DFG の例を示す．図には小規模な例を示したが，実際は幅と長さがより大きくなる．DEQ 1

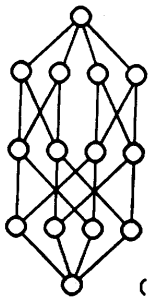

図 8 FFT モデルのデータフローグラフ
Fig. 8 Data flow graph of FFT model.

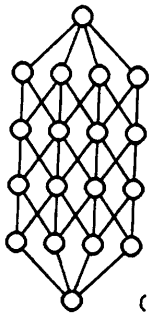

図 9 DEQ 1 モデルのデータフローグラフ
Fig. 9 Data flow graph of DEQ 1 model.

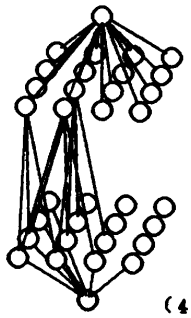

図 10 DEQ 2 モデルのデータフローグラフ
Fig. 10 Data flow graph of DEQ 2 model.

および DEQ 2 モデルは，緩和法による DFC 向きのアルゴリズムにおいて現われるループの 1 回分をモデル化したものである．

アクタの割付けには，DFG 中のアクタの番号順に各処理ノードに割り付けるという簡単なアルゴリズムを用いる．本システムは，分散形 DFC の割付け問題の困難さを軽減するため，分配回路網に，階層性や局所性の無い共通バスを採用している．したがって，各処理ノードはすべて同等であり，割付け問題はたんなる負荷配分の問題に帰着される．このため，規則的で固定的な DFG に対しては，上記のようなアルゴリズムでほぼ最適に近い割付けが得られる．

表 1 アクタの演算実行時間の測定結果
Table 1 Operation execution time of actors.

	演算回数	平均時間 (μsec)	分 散 (μsec²)	分布範囲 (%)
FET モデル	10	1,967	5,621	±6.6
DEQ 1 モデル	9	1,940	9,261	±8.6
DEQ 2 モデル	11	2,078	9,973	±8.3

試作システムの演算装置上で各 DFP のアクタの実際の実行時間を測定したところ，**表 1** に示す結果を得た．この演算装置の処理能力は 50 kFLOPS 程度であり，また，1 アクタ当りの浮動小数演算回数はいずれも 10 前後であるので，0.1 MFLOPS 程度のプロセッサを演算装置に用いれば，1 アクタ当りの平均実行時間は 100μsec となる．分布は 3〜4 個のピークをもつ比較的一様な形であった．分布範囲とは，一様分布を仮定した場合の，この分散値に対する分布の範囲である．

メッセージ長はすべて 32 バイトとする．ヘッダに 8 バイト程度を用いても，倍精度浮動小数データを 3 個含むことができる．共通バスのデータ線を 128 本と仮定しているので，1 メッセージは 2 バスサイクルで転送を終了する．

3. 提案システムの設計パラメータの導出

本章では，2 章で述べたモデルを用いたシミュレーション解析により，分散形 DFC の設計のための各種パラメータの導出を行う．

以下ではアクタの平均演算実行時間を α とおいて議論する．

3.1 メッセージ受信処理時間の導出

2.3 節の各 DFP モデルに対し，それぞれ処理ノード数 N が 32 と 64 の場合について，メッセージ受信処理時間 ρ をパラメータとして変化させながら，シミュレーションを行った．これらのうち，DEQ 2 モデルに対する演算装置と更新装置の利用率を**図 11** に示す．条件としては，受信処理時間の影響を避けるため，共通バスサイクル β を $2\times10^{-4}\alpha$ と十分高速にしている．また，アクタの演算実行時間は $\alpha\pm0.2\alpha$ の一様分布としている．

アクタの命令中に条件判断等があると，実行時間の分散が大きくなるが，固定的な数値計算等では表 1 にも示したように分散はあまり大きくならず，また，分布も問題やデータによりさまざまに変化するので，本シミュレーションでは上記のような条件を設定した．

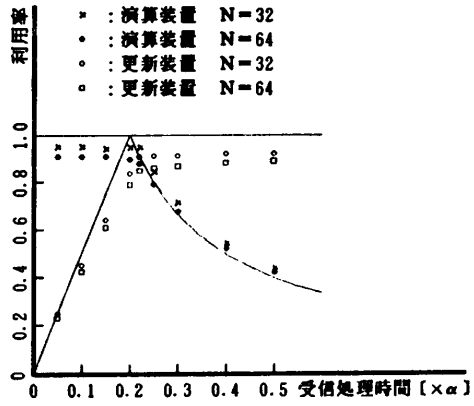

図 11 DEQ 2 モデルにおける受信処理時間に対する演算装置と更新装置の利用率
Fig. 11 Utilization ratio of operation unit and updating unit versus receiving process time on the DEQ 2 model.

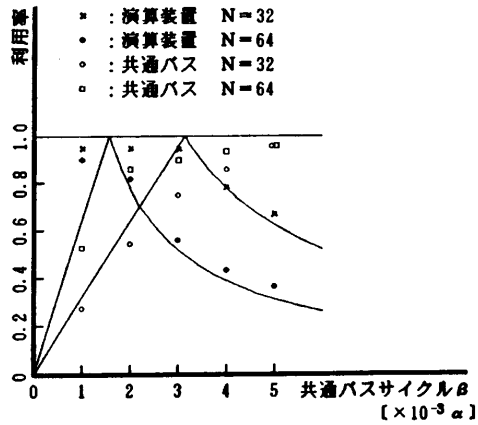

図 12 DEQ 2 モデルにおける共通バス速度に対する演算装置と共通バスの利用率
Fig. 12 Utilization ratio of operation unit and common bus versus common bus transmission rate on the DEQ 2 model.

DFP モデルの規模は，FFT モデルを 256×9, DEQ 1 モデルを 64×64，DEQ 2 モデルを 16×16×16 としている．したがって，並列度はそれぞれ 256，64，256 となる．

DEQ 2 モデルでは，1 アクタに対する入力リンク数が 5 であり，通常の数値計算としては，受信処理時間に対して厳しい条件となっている．図 11 より，演算装置の利用率を 0.8 以上とするためには，$\rho<\alpha/5$ にすればよい．他の 2 種のモデルについても同様のシミュレーションを行った結果から，並列度が処理ノード数の 4 倍程度以上ある DFP では，入力リンク数を λ とおいて，$\rho<\alpha/\lambda$ とすればよいと推定される．λ は通常の数値計算では 5 程度までであるので，提案システムの受信処理時間を 0.2α と設定し，以下ではこの値を用いて検討を行う．

3.2 共通バス速度の導出

共通バス速度を決定するため，2.3 節の各 DFP モデルに対し，処理ノード数が 32 と 64 の場合について，共通バスサイクル時間 β をパラメータとして変化させながらシミュレーションを行った．これらのうち，DEQ 2 モデルに対する演算装置と共通バスの利用率を**図 12** に示す．メッセージ受信処理時間を 0.2α，その他の条件は 3.1 節と同様とする．また，1 メッセージは 2 バスサイクルで転送を終了するものと仮定する．

受信処理時間と同様に，共通バス速度に対しても，DEQ 2 モデルは厳しい条件となっている．

図 12 より，演算装置の利用率を 0.8 以上とするためには，$\beta<0.133\alpha/N$ にすればよい．他の 2 種のモデルについても同様のシミュレーションを行った結果から，$\beta<2\alpha/3N\lambda$ とすればよいと推定される．したがって，処理ノード数が 256 の場合には，共通バスサイクル β を $0.5\times10^{-3}\alpha$ とすればよい．通常の TTL および MOS 技術を用いて 256 台の処理ノードを結合する場合，実現可能な共通バスサイクルは 50 nsec 程度であり[1]，これから，α の下限が 100μsec で抑えられる．

3.3 その他の各部の処理速度の導出

本節では，MF，DMAC，DMMU の処理速度に要求されるパラメータを導出する．

（1）メッセージフォーマッタの処理速度

MF は，アクタの実行結果のデータにヘッダを付加し，送信装置に転送する．したがって，メモリを 2 バンク切換えで構成すれば，最悪の場合でも 1 アクタ分の出力メッセージは次のアクタの実行中に転送を終了しなければならない．1 アクタ当りの出力メッセージ数を 5 とすると，1 メッセージを 0.2α で処理する必要がある．しかし，この処理時間は，そのままメッセージ遅延時間に加わるので，できるだけ短いことが望ましい．MF の制御は単純であり，ランダムロジックで組むことも可能である．試作システムでは，これを汎用のシーケンサとバイポーラ ROM を用いて実現しており，転送サイクルは 5 MHz である[1]．提案システムでは，MF のバッファの語長を 32 ビット構成とすれば，2.3 節に述べたように，メッセージ長は 32 バイトと仮定しているので，制御のための処理時間も含めて，1 メッセージ当り 2 μsec 程度の処理時

間の実現が可能である．したがって，$\alpha \gg 10\,\mu$sec ならば，メッセージ転送時間に対する MF の処理速度の影響は，実際上無視しうる．

（2）データメモリ転送装置の処理速度

DMAC は更新装置からの要求により受信装置から DM へメッセージのデータを転送する．更新装置が DMAC を起動するまでの処理時間は，メッセージ受信処理時間の約半分である[1]．その後の受信処理と転送は並行して行われるため，受信処理が終了するまでに転送を終了すればよい．受信処理時間を 0.2α と設定したので，1メッセージ分のデータ転送を 0.1α 以内で行えば，システム全体の性能に影響を与えない．DM を高速の RAM とアービタを用いて実現することにより，メモリのアクセス速度の 1/2 以上の転送速度が得られる[1]．DM の語長を 32 ビットで構成すると，メッセージ長を 32 バイトとしているので転送は 8 サイクルであり，2 μsec 程度の転送時間が実現でき，$\alpha > 20\,\mu$sec ならばオーバヘッドとはならない．

（3）データメモリ管理装置の処理速度

DMMU は，DM の空きブロックを管理し，メッセージ受信処理時に更新装置に空きブロック番号を1個与え，演算実行終了時に演算装置から使用済みのブロックを入力オペランド数だけ回収する．どちらも簡単なキュー操作で実現でき，両処理に必要とされる処理量も同程度である．各アクタの入力リンク数を5とすると，受信処理は α の間に平均5回起動され，演算装置は平均 α ごとに5個のブロックを解放するので，合計で α の間に 10 回程度のキュー操作が必要である．キューをハードウェアの FIFO で構成すれば，1回のキュー操作を 1 μsec 程度で行うことが可能であり，$\alpha > 10\,\mu$sec ならばオーバヘッドとはならない．

3.4 演算装置の演算実行時間に関する検討

以上に各部の処理に要求される処理速度と，実際的な速度について検討した．**表2**にこれらの結果をまとめた．要求処理時間とは，演算装置の平均実行時間を α としたときに各部に要求される処理時間，実際的処理時間とは，通常の TTL および MOS 技術で実現できる処理時間，下限演算実行時間とは，各部の要求処理時間と実際的処理時間から演算装置の演算実行時間の下限を逆算したものである．これらより，演算実行時間 α の下限は共通バスサイクルにより 100μsec に制限される．なお，この条件下での受信処理時間の実現性については文献 1)で検討している．

表 2 各部の処理時間
Table 2 Processing time of each process.

	要求処理時間	実際的処理時間 (nsec)	下限演算実行時間 (μsec)
受信処理	0.2α		
共通バスサイクル	$0.5\times10^{-3}\alpha$	50	$\alpha > 100$
MF転送	0.2α	2	$\alpha \gg 10$
DM転送	0.1α	2	$\alpha > 20$
DM管理処理	0.1α	1	$\alpha > 10$

3.5 各キューとデータメモリの容量の導出

メッセージ受信処理時間を 0.2α，共通バスサイクルを $0.5\times10^{-3}\alpha$ とし，その他の条件は 3.1 節と同様として，各 DFP モデルに対し処理ノード数 64，32，16 の場合についてシミュレーションを行い，各ノードにおける命令，送信，受信の各キューの長さの分布，および，データメモリに蓄積されているオペランドデータの数の分布を求めた．

FFT モデルにおける命令キュー長の確率分布を**図13**に示す．DFP モデルの並列度は 256 であるので，ノード数が 64，32，16 の場合の1ノード当りの並列度はそれぞれ 4，8，16 であり，命令キュー長がこれらの値より2だけ小さいところに分布のピークがある．他の2種の DFP モデルでは，入力リンク数がより多いので，更新装置の処理量も FFT モデルより多くなるため，更新装置がネックとなり，より命令キューが短いほうに分布が偏った結果を得た．これらより，命令キューには，1ノード当りの DFP の並列度と同程度の容量が望ましい．たとえば，256×256 の2次元微分方程式のような応用では，ノード数 256 として，1ノード当りの並列度は 256 となる．この場合，1アクタ当りの命令キューの大きさは，入力リンク数が5であることより，演算コードとオペランド所在情報を合わせて 12 バイト程度となる．したがって，キューの容量には3キロバイト程度必要となる．命令キューは比較的低速でよいため，この容量は実現可能

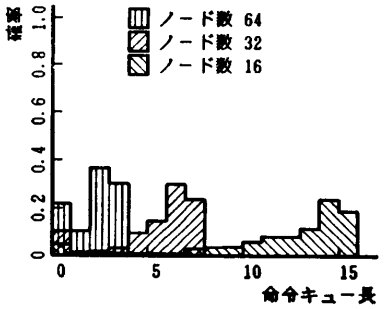

図 13 FFT モデルにおける命令キュー長の分布
Fig. 13 Distribution of instruction queue length on the FFT model.

な値である．より大規模な応用に対しては，更新装置内にソフトウェアキューを設ける等の処理が必要になる．ヘッダキューについても命令キューと同程度の容量が必要である[1]．

同様に，送信キュー長の確率分布を測定した結果，FFT モデルと DEQ1 モデルでは，キュー長が0である確率が0.9以上であり，4までに 0.98，16 までには 0.99 以上の確率で含まれていた．一方，DEQ2 モデルでは，共通バスがボトルネックになるため，送信キューにメッセージがたまる傾向にあり，1ノード当りの並列度と1アクタの出力メッセージ数の積の値近くまで分布が広がっていた．このように，共通バスがボトルネックにならない場合となる場合で，送信キュー長は著しく異なった分布を示す．現実には，後者の場合まで考慮して設計を行うことは困難である．前者の場合について，処理性能を低下させないよう，送信キューに空きがなくなる確率を 0.01 以下にするという条件を定めると，最低でも 16 メッセージ分必要である．共通バスがボトルネックになっている場合は，送信キュー長を制限しても，共通バスの使用率はほとんど変化しないので，全体としての処理速度にもほとんど影響がないと考えられる．

受信キューについても同様の測定を行った．最も入出力リンク数の多い DEQ2 モデルの場合の受信キュー長の確率分布を**図 14** に示す．ノード数が少ないほど分布はキュー長の短いほうに偏っている．すなわち，1ノード当りの DFP モデルの並列度は受信キュー長にはあまり関係なく，むしろ，ノード数が多くなるとメッセージの到着がポアソン到着に近くなることによりキュー長が長くなると考えられる．したがって，更新装置がボトルネックとならない限り，ノード数を256としても，受信キューには最低で10メッセージ分の容量があればよい．

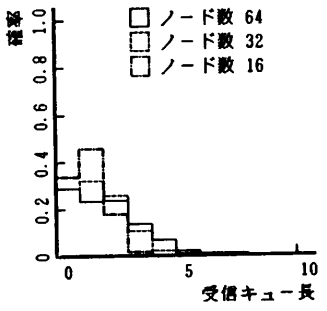

図 14 DEQ 2 モデルにおける受信キュー長の分布
Fig. 14 Distribution of receiving queue length on the DEQ 2 model.

表 3 各部に要求される容量
Table 3 Amount of storage required.

命令キュー	256 アクタ	3 kbyte
送信キュー	16 メッセージ	512 byte
受信キュー	10 メッセージ	320 byte
データメモリ	2,000 メッセージ	64 kbyte

DM に蓄積されるメッセージデータ数は，実行する DFP の1ノード当りの並列度と1メッセージ当りの入力メッセージ数に依存する．DM の容量が不足するとデッドロックに陥る可能性があるため，容量は十分大きくする必要がある．しかし，それでも不足する場合は，分割実行などにより DM の要求量を制限する必要がある．分割せずに直接実行できるプログラムの規模を，並列度を 10^5，1アクタ当りの入力メッセージ数を5と設定すると，1ノード当りの並列度は400程度となり，DM には最低で 2,000 メッセージ分の容量があればデッドロックは生じない．1メッセージを32バイトとすれば，各ノードに64キロバイト程度の DM を置けばよい．

以上の各部に要求される容量についての検討結果を**表3**に示す．

4. 提案システムの性能評価

演算装置の命令実行時間の下限は，共通バス容量により 100μsec に抑えられる．また，本稿で DFP モデルとして用いた応用は，いずれも1アクタ当り10個前後の浮動小数演算を含んでおり，最近のマイクロプロセッサが 0.1 MFLOPS 程度の処理能力をもっていることを考え合わせると，代表的な数値計算に関して，100μsec は1アクタの実行時間として妥当な値であると考えられる．この場合，演算装置の使用率は前述のように 80% 程度期待でき，256 ノード全体で 20 MFLOPS 程度の処理能力が得られる．

1アクタ当りの処理内容がこれより大きい場合は演算装置の使用率が100%に近づくことになり，システムの性能が十分引き出されるが，逆に小さい場合は共通バスおよび更新装置がボトルネックになり，性能が低下する．この場合，一般に処理内容が小さくなるとメッセージの大きさも小さくなる傾向にあると考えられ，共通バスの負荷は軽くなるが，受信処理時間はメッセージの大きさに関係なく一定であるので，更新装置がボトルネックになる．

本稿ではアクタの演算実行時間を $\alpha \pm 0.2\alpha$ の一様分布として評価を行ったが，命令キューに実行待ちの

アクタが存在している確率が高いので，分布や分散の変化は演算装置の使用率にあまり影響を及ぼさず，したがって，システム全体の処理能力への影響も少ないと考えられる．

5. むすび

本論文においては，すでに筆者らが提案している分散形データフロー計算機について，その設計のための諸パラメータの導出と，実現性の確認を目的として作成したシミュレータについて述べ，次にこれを用いて行ったシミュレーション結果から，提案システムの各部の処理速度や容量について検討した．

この結果，処理ノードの演算装置を 0.1 MFLOPS，共通バスサイクルを 50 nsec，メッセージ受信処理時間を 20μsec，処理ノード数を 256 としてシステムを構成することにより，アクタの処理内容を 10 浮動小数演算程度と仮定すると，システム全体で 20 MFLOPS の処理能力が得られることが明らかとなった．

これらは通常の MOS あるいは TTL 技術で実現可能であり，提案システムの実現性の確認が得られた．

謝辞 本研究に関し，種々の有益なご助言，ご討論をいただいた猪瀬・斉藤研究室の諸氏に感謝の意を表す．なお，本研究は文部省科学研究費の補助を得て行われたものであり，ここに記して謝意を表す．

参考文献

1) 大山，グェン，スレスタ，斉藤，猪瀬：分散形データフロー計算機の構成とその実験的評価，情報処理学会論文誌，Vol. 25, No. 1, pp. 101-108 (1984).
2) 浅田，グェン，堀，斉藤，猪瀬：データフロー計算機における故障検出の一方式，情報処理学会分散処理システム研究会資料，1-4, pp. 1-10 (1979).
3) グェン，浅田，大山，斉藤，猪瀬：環状回路網を用いた分散型データフロー計算機の一方式，電子通信学会論文誌，Vol. J 65-D, No. 12, pp. 1528-1535 (1982).

(昭和 58 年 10 月 12 日受付)
(昭和 59 年 4 月 17 日採録)