

機能メモリ主導型データ駆動アーキテクチャQ-FM とその評価

岩田 誠 寺田 浩詔

E-mail: {iwata, terada}@ise.eng.osaka-u.ac.jp

大阪大学 工学部 情報システム工学科

〒565 吹田市山田丘 2-1

計算物理や画像処理に類出する大規模セル型データ処理は膨大な並列性を有し、大量の計算能力と記憶容量を要求する。本論文は、このような大規模セル型データ処理を超並列に実現するために、遅延耐性に優れた動的データ駆動原理を採用した、機能メモリ主導型データ駆動アーキテクチャ *Q-FM* を提案している。*Q-FM* は、動的データ駆動型プロセッサとその処理進行状況に適応してデータを効果的に供給する機能メモリ機構からなる、受動的アーキテクチャを採用している。実験評価の結果、*Q-FM* では、メモリアクセス遅延時間とは無関係に、プロセッサの稼働率を最大に維持できることを確認した。さらに、本アーキテクチャをマルチプロセッサに適用すれば、スケーラブルに処理能力を向上できることも実験的に確認した。

A Dedicated Data-Driven Architecture Q-FM Coordinated by Functional Memory

Makoto IWATA and Hiroaki TERADA

Department of Information Systems Engineering, Faculty of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka 565, Japan

A class of large-scale cell-data processing problems in computational physics and image processing applications requires enormous parallel processing capability and storage capacity. This paper proposes a dedicated data-driven architecture Q-FM coordinated by functional memory to process the data-stream parallelism inherent in most of the cell-data processings. In order to utilize latency tolerance feature of the dynamic data-driven scheme, Q-FM employs a passive architecture in which a functional memory mechanism actively and adaptively supplies stored data to a data-driven processor. Experimental simulation results show Q-FM preserves maximum efficiency of the processor independent of memory access latency and achieves scalable performance gain on data-driven multiprocessor system.

1 はじめに

実用的な超並列処理システムの構築には、ソフトウェア、アーキテクチャ、ハードウェアそれぞれの要請を総合的に捉えることが重要である。すなわち、アーキテクチャを意識せず、容易にプログラム可能であり、かつ、各種の遅延に対する耐性を持つ要素プロセッサを、隘路を生じないネットワークにより多数接続し、スケーラブルに性能を向上できなければならない [1]。さらに、高集積化が可能で、電力消費が少ないハードウェア実現法が要求される。

筆者らはこれまで、これらの要請を原理的に満足させるため、自然なシステム表現能力を有し、かつ、細粒度から粗粒度にいたる並列処理が可能な動的データ駆動原理を採用したアーキテクチャを、極限集積化向きの自己同期型緩衝パイプライン機構によりハードウェア実現し、数種のスーパーパイプライン型プロセッサ $Qv-x$ を開発してきた [2, 3]。

本論文では、これらの研究成果を基礎とし、超並列処理の応用プログラムの大半を占めると予想される、大規模なセル型データ処理を効果的にパイプライン処理する、機能メモリ主導型データ駆動アーキテクチャ $Q-FM$ (Data-Driven Processor Q coordinated by Functional Memory) を提案する。 $Q-FM$ が対象とする大規模なセル型データの処理は、計算物理における解析空間データや画像処理における画像データなどを対象とする、高い並列計算能力と大量の記憶容量を要求する処理である。

現状の計算物理や画像処理などの分野では、ベクトル計算機やノイマン型のいわゆる超並列マシンが多く利用されている [4, 5]。しかし、前者は、問題が複雑になると、DO ループ内の処理に逐次代入や IF 文などを含ませないようベクトル化する作業が困難になるという問題がある。また、後者に対しては、大規模な並列処理を効果的に実現するために、遅延時間の低減・隠蔽方式の提案・評価が各所で行われている [6, 7, 8]。並列ベクトル計算機でも同様に、これら両者の課題を併せ持っている。

これに対して、動的データ駆動型処理方式では、問題の持つ多次元的な並列性が自然に図的に表現できる [9] ため、プログラムから並列性を抽出する作業が不要である。さらに、この方式では、データの到着によってのみ処理が駆動され、一旦発火した処理は決して中止されず、パイプラインの制御が非常に簡単になるので、多段のパイプライン処理が可能である。このため、メモリアクセスやプロセッサ間通信などの種々の遅延はシステム全体の処理率 (単

位時間あたりの処理量) に影響しないという原理的な特徴がある [1]。

本稿では、これらのデータ駆動型処理方式の特徴に着目して、各種の遅延時間とは無関係にプロセッサ稼働率を最大限に維持可能な、機能メモリ主導型のデータ駆動アーキテクチャ $Q-FM$ の構成法について述べる。以下次章では、計算物理や画像処理に頻出するセル型データ構造に対する処理形態を並列処理性の観点から考察し、並列処理性を余すところなく実現可能なデータ駆動アーキテクチャ $Q-FM$ を提案する。3 章では、 $Q-FM$ による分散メモリ型マルチプロセッサの構成法を提案し、4 章では、我々が開発した汎用のデータ駆動型プロセッサ $Qv-x$ [2, 3] 上での $Q-FM$ のシミュレーション実験の評価結果を示し、メモリアクセス遅延と無関係にスケーラブルに処理能力を向上できることを示す。

2 大規模セル型データ処理の並列処理性

本論文で対象とする大規模セル型データ処理とは、多次元座標系内の各セルに対応づけられたデータの各々に対して、その近傍データを含めて均質な計算を施す処理形態を持つものである。例えば、計算物理の分野での、分布定数系の数値シミュレーションは、解析対象空間を多数の格子に分割して、各格子について連立微分方程式を解く。また、画像処理の分野では、一般に各画素についてその近傍画素を含めた変換処理が頻出する。

これらの処理では、実用的な精度を得るために、多次元座標系の最小単位を細かくする必要があり、大規模なセル・データに対する大量の計算能力と記憶容量が要求される。実用的規模のセル型データは最終的には線形アドレス空間を持つ低速・大容量記憶装置に格納せざるを得ない。このため、大規模セル型データ処理の効果的な実現の鍵は、メモリアクセス遅延時間の影響の克服して、問題の持つ並列処理性を余すところなく効果的に実行する方式にあると言える。

したがって、まず、セル型データ処理を並列処理性の観点から、次の属性について考察してみる (図 1 参照)。

- (1) 入力データ 各セルの計算に必要なデータ集合 (対象セルのデータののみ/近傍セルのデータ)
- (2) データ形式 各セルに割り当てられるデータの構造 (スカラー/集合)
- (3) 依存関係 各セルの計算間のデータ依存関係 (なし/座標軸方向)

(4) セル空間形状 計算対象にする座標空間 (固定／実行時に動的に変化)

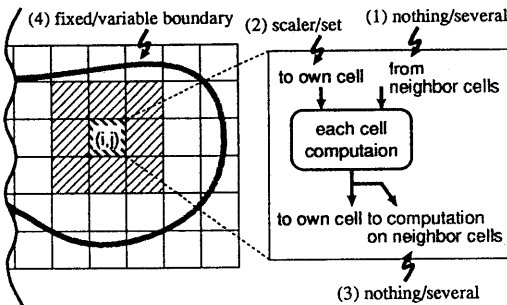


図 1: セル型データ処理の概要と特徴

これらの属性から一般に、大規模セル型データ処理に内在する並列処理性は、次のように特徴づけられる。

- (a) セル間のデータ依存関係が隣接セル間に限定できるため、問題空間を部分空間に分割して並列に実行可能である。…粗粒度の並列性
- (b) 各セルに関する計算をセル間のデータ依存関係に従ってパイプライン並列に実行できる。…中粒度の並列性
- (c) 各セルに関する計算は、隣接セルの複数のデータを入力とする計算グラフとして一般化でき、算術演算プリミティブレベルの並列処理性がある。…細粒度の並列性

すなわち、(a) はマルチプロセッサ水準で、(b)(c) はプロセッサ水準での並列処理によって活用可能である。

3 Q-FM の基本アーキテクチャ

セル型データ処理では一般に、前章に述べたように、各セルの計算をパイプライン並列に実行できる。このため、メモリアクセス遅延時間と無関係に、記憶装置側が能動的に計算に必要な大量のデータ流を組織的にプロセッサ側に供給するシステム構成が採用できる。従って Q-FM は、各種のセル型データ処理アルゴリズムの先行関係とその実行の進捗に応じてデータを適応的に供給できる機能メモリ機構と、ストリーム処理能力に優れた動的データ駆動プロセッサ $Qv-x$ とから構成されている。以下本章では、Q-FM の全体構成を示し、各種のセル型データ処理に適用可能な機能メモリの構成法を提案する。

3.1 データ供給機構を伴う受動的アーキテクチャ

データ駆動型処理方式は、細粒度の同時並行・パイプライン並列処理能力を有するため、対象とするアルゴリズムの並列性に依拠して遅延耐性を発揮する。このため、大規模マルチプロセッサ化の際にも、原理的にスケーラブルな性能向上が見込める。大規模なセル型データ処理においても、このような特性を発揮できるシステムを実現するには、各々のセルに関する計算ができる限り並列に動作できるよう、メモリからデータストリームを適切な順に供給し、データ駆動原理の受動的性質を活用したパイプライン並列処理を究極まで追求するのが効果的である。

したがって、本論文で提案する機能メモリ主導データ駆動アーキテクチャ Q-FM では、図 2 に示すように、データ駆動型プロセッサとこのプロセッサにデータを供給する機能メモリ機構からなる受動的なアーキテクチャの構成を採用した。この機能メモリは、後述するように、比較的簡単なハードウェア機構により実現可能であり、プロセッサの並列処理能力に見合うようにデータ供給レートを実行時に制御することによって、メモリアクセス遅延時間と無関係にプロセッサ稼働率を常時最大に維持することを可能にする。

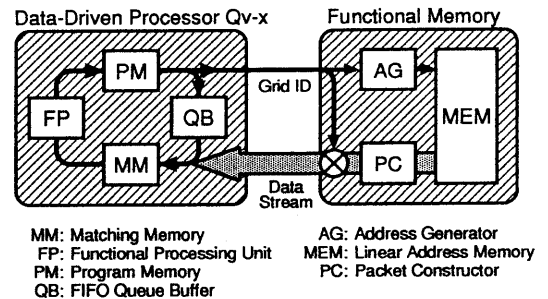


図 2: 機能メモリ主導型データ駆動アーキテクチャ Q-FM の機能構成

この Q-FM のデータ駆動型プロセッサには、筆者らが共同開発したスーパーパイプライン型の $Qv-x$ プロセッサ [2, 3] を用いている。この $Qv-x$ のハードウェア実現は、図 2 の左部に示されるように、

- データ駆動型処理実行に必要な機能要素（待ち合わせ記憶部 (MM)、関数的処理部 (FP)、プログラム記憶部 (PM)）と FIFO 型緩衝記憶 (QB) を、すべて多段のパイプライン型処理機構として実現し、さらに、

- これらの機能要素をふたたびパイプライン型に環状接続した、

スーパーパイプライン型のシステム構成として設計されている。これらのパイプライン型処理機構はすべて自己同期型データ転送・処理機構により実現され、大域的なシステムクロックやバスを排除した、完全な自律分散制御がハードウェア水準で実現されている。このため、従来のノイマン型アーキテクチャの大規模高集積化に伴うクロックスキューの問題や配線長増大の問題が原理的に解決されている。

このプロセッサに組織的にデータ流を供給するために、機能メモリは図2に示すように、線形アドレス記憶内の大規模セル型データ構造への各種のアクセス形態に応じてアドレスを生成する機構 (AG)、ならびに、アクセス結果データにタグを付与してパケット化する機構 (PC) から構成されている。アドレス生成機構 (AG) は、次節以降に述べるように、静的スケジューリングにより規定された命令に基づき動作する。同時に、パケット構成機構 (PC) から生成されたデータ流は、適応的データ供給機能により、プロセッサ側の処理進行に応じて柔軟に供給される。

3.2 静的スケジューリング機能

セル型データ処理においては、各セルに関する計算が独立に実行可能な場合には、マクロな計算間のデータ依存関係を満足してさえいれば、各格子の計算順序は任意にできる。

一方、各セルに関する計算間にデータ依存関係がある場合、例えば、各セルの先行関係が図3に示すように、各次元の軸方向にある場合には、対角線方向のセル群に関しては同時並行に実行できる。従って、対角線方向に各格子の計算を起動可能なように、予めスケジューリングする機構を用意すれば、セル型データ処理の持つ中粒度の並列処理性の活用が可能になる (図3参照)。

このようなスケジューリングを可能にするために、Q-FMでは、表1の命令セットとこれを実現する図4の機能メモリ機構を有している。

表中の近傍情報 (neighbor) は、必要な近傍セルのビットパターンとして与えられる。これは、(隣接近傍を扱う場合) 解析対象空間が2次元の場合9bit、3次元の場合27bitで表現される。機能メモリ機構内のアドレス生成機構 AG (Address Generator) では、命令パケット中の開始アドレス (base_address) と走査順序一覧テーブル (scanTBL_ID) を元に、計算対象セルのデータを走査するためのアドレスを計算し、上記

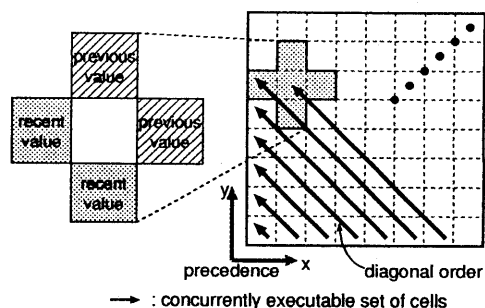


図3: 先行関係を満足する静的スケジューリング法

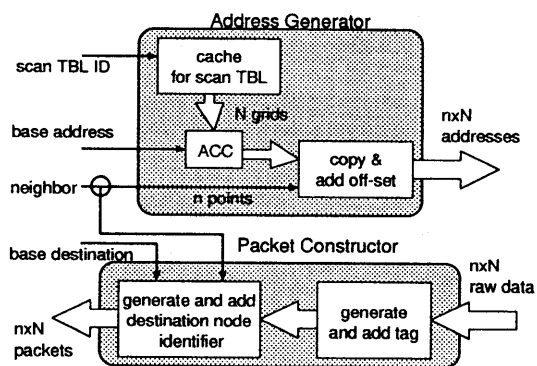


図4: 機能メモリ機構の機能構成

の neighbor から近傍データのアドレスを生成する。また、パケット構成機構 PC (Packet Constructor) では、メモリへのアクセス結果データに、neighbor と base_destination (行き先ノード ID の初期値) から求めたタグを付与して、データ駆動プロセッサで処理可能なパケット形式を構成する。

なお、本機能メモリ機構は、セル空間形状が複雑な場合にも、その空間形状に対応した走査 TBL をメモリに記憶し、実行時にキャッシュに置くことにより、高速アクセスを実現できる。これによって、任意の空間形状に対して、データ供給率を高く維持できる構成を採っている。

3.3 適応的データ供給機能

セル型データ処理では、セル空間の形状に応じて境界条件を加味して計算する必要がある場合などのように、各セルに関する計算の負荷は必ずしも一定にはならない。従って、実行時に処理進行状況、すなわち、次に供給すべきデータが受理可能かどうか

表 1: 機能メモリへのアクセス命令一覧

命令名	個別パラメータ	機能
s.read		scanTBL に従って、データを連続的に参照
s.write	data	scanTBL に従って、データを連続的に更新
ns.read	neighbor	scanTBL に従って、近傍データを連続的に参照
ns.write	neighbor, data	scanTBL に従って、近傍データを連続的に更新

†共通パラメータ: base_address, scanTBL_ID, base_destination

を機能メモリ機構にフィードバックして、データ供給率を適応的に制御する必要がある。このような制御を行う方式として、以下の2つの方式を提案する。

(a). 各格子内の処理進行イベントによる方式

適応的なデータ供給手法として、プロセッサから現在実行中のセル番号を機能メモリに随時フィードバックし、予め静的に決定されたスケジュールをこのフィードバック情報を元に進行させる。本機能は、図2右部の⊗で示す箇所、実現される。ハードウェア的には、この箇所の自己同期型パイプライン段の制御回路にNACKを発生する回路を付加して、容易に実現可能である。

(b). QB 内の滞留トークン数による方式

プロセッサの負荷状況を、 $Qv-x$ プロセッサのFIFO型緩衝記憶QB内に滞留しているバケット数から判断する。 $Qv-x$ は、自己タイミング方式によりハードウェア実現されているため、バケットの滞留数が非常に簡単に検出でき、上記(a)の回路によりデータ受付を容易に制御できる。本方式では、二つの閾値 T_{low} 、 T_{high} を設定して、一種のヒステリシス特性を付与して、負荷状況をフィードバックする。すなわち、QB内の滞留バケット数が T_{high} を越えた時、データ供給を停止し、 T_{low} 未満になった時、データ供給を再開する。

以上のように、 $Q-FM$ アーキテクチャは、静的スケジュールリング機能と適応的なデータ供給機能を合わせ持つ、機能メモリ機構が主導して、データ駆動型処理を適応的に進行可能なため、定常状態になれば、メモリアクセスの遅延時間に関係なく、データ駆動型プロセッサを最大限に稼働できる特徴を有している。

4 $Q-FM$ マルチプロセッサ

本章では、セル型データ処理の持つ粗粒度の並列性を活用するために、前章に述べた $Q-FM$ アーキテクチャに基づいて、マルチプロセッサ・システムの実現法を提案する。このためには、各プロセッサへの計算負荷の割り当てとデータの分割・配置手法を考慮したシステム構成ならびにプロセッサ間通信方式が重要になる。

以下では、まず、 $Q-FM$ によるマルチプロセッサ・システムの構成の概要を説明し、本システムに採用した、データの分割・配置手法、緩やかなプロセッサ間同期方式、ならびに、アドレス計算を簡単化可能なメモリマッピング方式について述べる。

4.1 $Q-FM$ によるマルチプロセッサ構成

計算力学や画像変換などのセル型データ処理では、各セルの計算に必要なデータは近傍セルのデータだけであるため、プロセッサ毎にローカルなメモリが付与された分散メモリ型マルチプロセッサ構成を採用できる。また、構造化されたセル空間をそのまま写像可能なメッシュ型プロセッサ間接続トポロジを採った(図5参照)。さらに、プロセッサ間の通信が、通信量や受け側プロセッサの負荷に応じて、柔軟に実現されるように、エラスティック・パイプライン機構によりハードウェア実現されたルータチップ[10]を採用した。

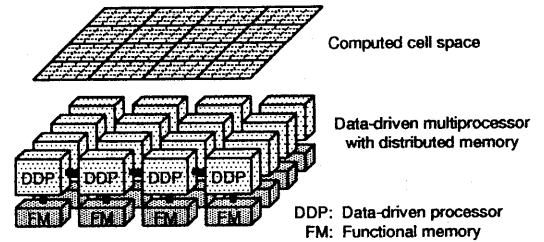


図 5: データ駆動型マルチプロセッサ上へのセル空間の分割・配置の一例

4.2 データの分割配置手法

各プロセッサには、図6に示すように、処理対象のセル空間の部分空間に対する計算プログラムを割り当て、部分空間に属するセルのデータをローカルな機能メモリに配置する。セル型データ処理では、近傍セルを対象に計算が行われることが多いため、各セルのデータの参照頻度は更新頻度より多い。このため、分割した領域の境界に位置するセルのデータ

に関しては、隣接プロセッサ間で重複して記憶し、円滑なデータ供給が可能になるようにした。また、データ更新は、遠隔メモリアクセス命令により、直接的に隣接プロセッサ内のセル型データ構造を更新する操作によって実現される。これらの手法によって、遠隔のセル型データ構造の参照を局所的なデータ構造への参照により代替できるため、メモリアクセスの遅延時間の短縮が可能となり、さらに、プロセッサ間通信量が極小化できる。

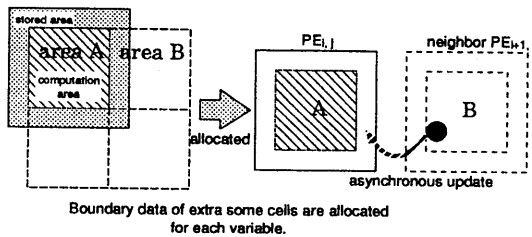


図 6: データの重複分割・配置手法

5 Qv-x上での実験・評価

Q-FMの持つ遅延耐性とスケーラビリティの実験的評価のため、2章に述べたセル型データ処理の属性を考慮した一般的な問題として、流体シミュレーション手法の典型である SIMPLE(Semi-Implicit Method for Pressure Linked Equations) [11] 法を取り上げた。本プログラムでは、連立した離散化方程式の求解には、各セルの計算間に対角線方向の依存関係がある、逐次過緩和法 (SOR) を用いた。

5.1 データ供給制御方式の性能評価

従来のノイマン型プロセッサの遅延隠蔽方式の代表例に、多重スレッド実行とプリフェッチがある。これらを併用した方式を提案方式の比較評価のために用いた。データ駆動方式では、本質的に多重スレッドやプリフェッチの概念は存在しないので、細粒度の並列実行、ならびに、直接的なデータ依存関係がある命令間へのデータ読み出し要求の挿入によって、疑似的に実現し、プロセッサの ALU 稼働効率を測定した。

図 7は、SIMPLE 法の内、最も負荷が大きい速度近似計算のみを抜粋して、メモリアクセス遅延時間の影響を現行の Qv-x プロセッサの Version 2m (通称 RAPID[3]) 上で評価した結果である。具体的には、機能メモリ機構をシミュレートするデータ駆動型プログラムと、流体計算を行うデータ駆動型プロ

グラムを、それぞれ、別々のプロセッサ上に割り当て、Q-FMの動作を模擬した。同様の手法で、上述のプリフェッチ方式の動作も模擬した。

図 7中のプロセッサ稼働効率 E(Efficiency) は、与えられたプログラムに対する理論的最大性能 R_{ideal} [12] に対する、実測値 R_{real} の比率で与えられている。

$$E = \frac{R_{real}}{R_{ideal}} = \frac{R_{real}}{R_{max}(I_u + I_b)/(I_u + 2I_b)}$$

ここで、 I_u 、 I_b はそれぞれ、実行される単項演算命令数、二項演算命令数を示し、 R_{ideal} は、二項演算命令の実行により減少するパケット流量分を最大パケット流量 R_{max} に乘じた値である。

図 7から容易にわかるように、プリフェッチ方式では、メモリアクセス遅延時間の増大に伴って、ALU 稼働効率が低下する。また、データ先読みが深ければ、遅延時間の隠蔽効果は高いが、遅延時間が短い場合にはデータ供給過多になり実行不能になる。これに対して、Q-FMでは、メモリアクセス遅延時間に無関係に、ほぼ最大の性能を発揮できることが分かる。ただし、この実験では、3.3 節に述べた、QB 内の滞留トークン数によるハードウェア制御を行っていないため、数%の性能劣化が起きている。

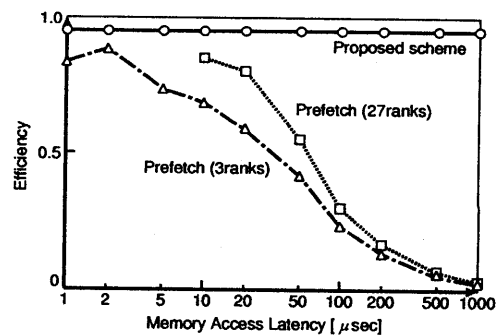


図 7: 遅延時間に対するプロセッサの稼働率

また、単位時間あたりに計算可能な格子数を各種の RISC プロセッサと比較した結果を表 2 に示す。動的データ駆動型プロセッサ Monsoon の性能は、文献 [13] による。SPARC と Alpha に関しては、100 × 100 格子と 1000 × 1000 のキャビティ流れの計算 (100 回反復) をそれぞれ実行して、UNIX の time コマンドにより測定した結果である。Qv-x はクロックを持たないため、これらの結果と単純には比較でき

ないが、表 2 では、パイプライン・タクトの逆数を取り、50MHz 相当として、比較している。この表から、*Q-FM* が高い性能を達成し、さらに、格子空間の規模が増大しても、計算性能が維持できる能力を持つことが判る。

表 2: 種々のマシン上での SIMPLE 法の計算性能比較

system	[MHz]	external cache	MEM	grid space	
				10K†	1M†
Monsoon[13]	(10)	0	—	2.1 K	—
microSPARC (85)		0	32MB	28 K	19 K
superSPARC (60)		1MB	96MB	56 K	53 K
alpha 21064 (150)		256KB	64MB	52 K	49 K
<i>Q-FM</i> (≈ 50)		0	16MB	59 K	59 K

†[GPS]: calculatable grids per seconds

5.2 マルチプロセッサシステム上での性能評価

SOR 法を利用した SIMPLE 法に従った純粋な計算のみを実行するプログラムを対象として、これに、メモリから供給されるべきデータを外部から入力することによって、本方式の性能を RAPID シミュレータにより間接的に測定した。

データ供給率を一定にした場合と適応的に変化させた場合について、マルチプロセッサ上での並列処理効率を測定した。プロセッサ 1 台あたりに割り当てる格子数は均一にし、動的な負荷分散は行っていない。この際、SIMPLE 法における各速度成分、圧力、補正の部分計算毎に、最適な入力間隔でデータを投入することによって、上記の適応的データ供給方式を近似して、実験を行った。

図 8 は、プロセッサあたりに割り当てる格子数が 64×64 の場合に、単位時間あたりに計算可能な格子数を、マルチプロセッサによる並列処理効果として、示したものである。この結果から分かるように、適応的スケジューリング機能を持つ機能メモリ機構を持つ *Q-FM* マルチプロセッサでは、(1) 各プロセッサのストリーム並列処理能力によって、メモリアクセス遅延時間と無関係にプロセッサ稼働率を最大限に維持でき、かつ、(2) 大規模セル型データを部分空間に分割しても、隣接プロセッサ間のみの通信によって同時並行に処理を実行可能なため、プロセッサ数の増強に応じて、線形な性能向上を達成できる。

また、この実験から、本方式では、プロセッサ間通信流量が、最大でも 240 Kpackets/sec (42 ビット構成のパケット) 程度であることがわかった。こ

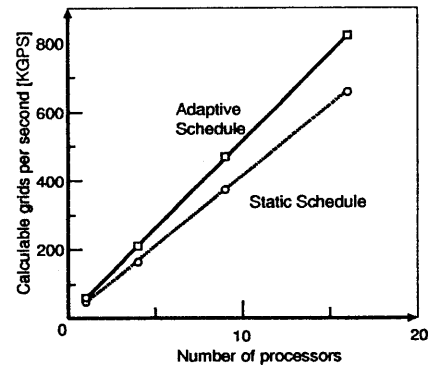


図 8: マルチプロセッサによる並列処理効果

れは、比較的低い通信容量のルータでも、容易に *Q-FM* のマルチプロセッサ化が可能であることを示唆している。

6 おわりに

本稿では、データ駆動原理の受動的性質を活用して、計算物理や画像処理に頻出する大規模なセル型データに対する処理を効果的に並列実現する専用アーキテクチャ *Q-FM* として、動的データ駆動型プロセッサと、これに適応的に高速データ・ストリームを供給する機能メモリ機構とから構成されるアーキテクチャを提案した。実験の評価結果を通して、本方式が、大規模なセル型データ処理を、メモリアクセス遅延時間と無関係に効果的に並列実現可能なことを示した。

従来のノイマン型アーキテクチャは、原理的に遅延時間に対する耐性がないため、問題規模の増大に伴って、メモリアクセス遅延が増え、プロセッサの処理率が劣化する。本論文で提案したデータ駆動型処理方式による機能メモリ主導アーキテクチャ *Q-FM* は、これらの問題を原理的に解決して、スケラブルな超並列処理システムの実現を可能にする一つの方法になり得る。

今後の課題としては、現状の *Qv-x* エミュレータ上では評価不可能な、QB 内の滞留パケット数による適応的データ供給方式の効果を検証するために、提案した機能メモリ機構のハードウェア実現とその実証的評価を行う必要がある。また、本アーキテクチャを大規模流体計算や画像処理への適用が残されている。

本論文の冒頭でも簡単に触れたが、データ駆動型

ソフトウェアは、問題やその解法の多次元的な構造を自然に表現できる特徴がある。このため、プログラム表現に密着した、アルゴリズム・アニメーションやシミュレーション結果の視覚化が可能であるという特徴がある [14, 15]。この特徴を活用すれば、*Q-FM* をエンジンとした、大規模セル型データ処理の統合的な環境を提供するシステムが実現できる可能性がある。これに関しては、稿を改めて論じたい。

謝辞 御指導、御支援頂いた *Qv-x* 研究の関係各位ならびに御協力頂いた寺田研究室の各位に深く感謝する。特に、本論文の実験結果の一部は、(現 JR 西日本) 山野弘喜君が本学修士課程在学中に行った研究による。また、データ駆動型プロセッサ *RAPID* に関して、日頃から御議論いただく三菱電機(株)の田村俊之氏・坪田浩乃女史に感謝する。

なお、本研究の一部は、文部省科学研究費(一般 B-2 05452363, 試験 B-1 06555110)の援助を受けて行なったものである。

参考文献

- [1] Arvind and R.Iannucci: "Two Fundamental Issues in Multiprocessing", Proc. 10th Int'l Symp. on Computer Architecture, pp.426-436 (June 1983).
- [2] Terada H., Iwata M., Komori S. and Miyata S.: "Superpipelined Dynamic Data-Driven VLSI Processors", In Gaudiot J.L., Bic L. and Gao G.R., eds., Advanced Topics in Dataflow Computing and Multithreading, IEEE Computer Society Press (Mar. 1995).
- [3] Komori S., Tamura T., Asai F., Tsubota H., Sato H., Takata H., Seguchi Y., Ohno T., Tokuda T. and Terada H.: "A 50 MFLOPS Superpipelined Data-Driven Microprocessor", Proc. Int'l Solid State Circuit Conference '91, pp.92-93,294 (Feb. 1991).
- [4] Simon H.D. ed.: "Parallel Computational Fluid Flow", p.345, MIT Press (1992).
- [5] Tezduyar T., Aliabadi S., Behr M., Johnson A. and Mittal S.: "Parallel Finite-Element Computation of 3D Flows", IEEE Computer, **26**, 10, pp.27-36 (Oct. 1993).
- [6] Gupta A., Hennessy J., Gharachorloo K., Mowery T. and Weber W.D.: "Comparative Evaluation of Latency Reducing and Tolerating Techniques", Proc. 18th Int'l Symp. on Computer Architecture, pp.254-263 (May 1991).
- [7] 平木, 島田, 関口: "細粒度並列処理におけるレイテンシ隠蔽効果の評価", 情処学論, **35**, 4, pp.619-627 (Apr. 1994).
- [8] Chen T.F. and Baer J.L.: "A Performance Study of Software and Hardware Data Prefetching Schemes", Proc. 21st Int'l Symp. on Computer Architecture, pp.223-232 (Apr. 1994).
- [9] A.L.Davis and R.M.Keller: "Data Flow Program Graphs", IEEE Computer, **15**, 2, pp.26-41 (Feb. 1982).
- [10] Seguchi Y., Komori S., Takata H., Tamura T., Asai F., Sato H., Tokuda T. and Terada H.: "A Flexible Router Chip for Massively Parallel Data-Driven Computer", Proc. IEEE Symposium on VLSI Circuits, pp.27-28 (May 1991).
- [11] Patankar S.V.: "Numerical Heat Transfer and Fluid Flow", p.197, Hemisphere Publishing Co.(1980).
- [12] 坪田, 田村, 高田, 浅井, 佐藤, 瀬口, 小守, 寺田: "シングルチップデータ駆動型プロセッサのアーキテクチャ評価", 情処学論, **34**, 4, pp.606-615 (Apr. 1993).
- [13] Hicks J., Chiou D., Ang B.S. and Arvind: "Performance Studies of Id on the Monsoon Dataflow System", MIT technical report, No.345 (May 1993).
- [14] 岩田, 寺田: "図的仕様記述からのデータ駆動型プログラムの生成手法", 情処学論, **36**, 5, pp.1203-1210 (May 1995).
- [15] 岩田, 坪田, 田村, 小守, 寺田: "動的データ駆動型処理システム *Qv-x* の視覚的評価支援環境", 信学技報, *CPSY94-79*, pp.57-64 (1994-12).