1H−1

MIMD型並列計算機システムを指向したメッセージフロー型並列計算モデル

藤井啓明，柴山 潔
(京都大学工学部)

1 はじめに

並列計算機システムのアーキテクチャを設計する場合，そのシステムに適用する計算モデルが重要な鍵となる．

我々は種々のシミュレーションなどの応用分野に適用可能なMIMD型並列計算機システムの開発[1]を進めている．このシステムは数千台規模の要素プロセッサ（PE）を持ち，PEには関数実行レベルの粒度（単位処理機能）を割り当てる．

このような並列計算機システムを指向した並列計算モデルとしては，これまでに，i) マクロ・データフロー・モデル; ii) アクター・モデルおよびその発展形である並列オブジェクト指向モデル; などが提案されている．

しかし，これらの計算モデルにはそれぞれ，i) 並列計算機システム上で扱う問題を表現するのが難しい; ii) そのモデルの並列計算機上へのマッピングが難しい; という問題が存在していた．

我々は，従来の並列計算モデルが持つこれらの問題点を解決することを目標とした新しい並列計算モデルである「メッセージフロー型並列計算モデル」(以降,「メッセージフロー・モデル」と略称する）の提案を行なう．

2 メッセージフロー型並列計算モデル

メッセージフロー・モデルは，並列オブジェクト指向モデルに基づき，これに，マクロ・データフロー・モデルの概念を融合したものである．

並列オブジェクト指向モデルには，i) 応用分野における対象のモデル化が容易である; ii) MIMD型並列計算機システムに対して表1に示すような関連付けが可能であり，モデルのシステムへのマッピングが比較的容易である; といった利点が存在するが，ii) に関しては，そのマッピングが必ずしも効率の良い並列処理にはつながらないという問題がある．この問題の一因は，並列オブジェクト指向モデルにおいて，特定のオブジェクトに対して送られるメッセージ間のスケジューリングや，異なる複数の処理の流れに対する同期に関してそれほど注意が払われておらず，それに対する強力な機構が装備されていないことである．

そこで我々は，並列オブジェクト指向モデルが持つこのような欠点を解決するために，データフロー・モデルにおけるデータの待ち合わせという強力な同期機構を応用した並列計算モデルとして，メッセージフロー・モデルを提案する．すなわち，メッセージフロー・モデルにおいて中心となる考え方は,「並列オブジェクト指向モデルでのメッセージの流れをマクロ・データフロー・モデルにおけるデータの流れに同化する」ということである．

並列オブジェクト指向モデル	並列計算機システム
オブジェクト	PE
メソッドの並列実行	MIMD方式
情報隠蔽	分散メモリ
メッセージ送受信	PE間通信

表1: 並列オブジェクト指向モデルとMIMD型並列計算機システムの関連付け

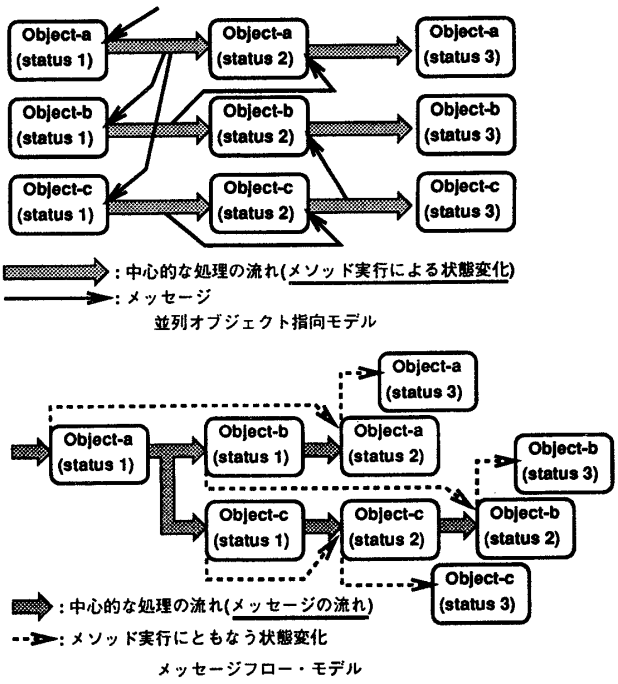

図1: メッセージフロー・モデルと並列オブジェクト指向モデルの比較

図1に示すとおり，従来の並列オブジェクト指向モデルでは，一連のメッセージを流れとしてはとらえていない．すなわち，このモデルに基づく処理の中心的な流れはあくまでも複数個のオブジェクトにおける個々の状態変化やそのための活動であった．これに対して，メッセージフロー・モデルでは，一連のメッセージの組を処理の流れとしてとらえ，これをこのモデルに基づく処理の中心的な流れとする．そして，このメッセージの流れをデータフロー解析的な手段で解析することによって，並列オブジェクト指向モデルの問題点の一つであった同期機構の弱さを克服することを可能としている．

なお，このデータフロー解析的な手段のことをメッセージフロー解析と呼ぶ．さらに，図1中のメッセージフロー・

Message-flow: A New Computation Model for MIMD-type Parallel Machines
Hiroaki FUJII, Kiyoshi SHIBAYAMA
Kyoto University

モデルにおけるメッセージの流れは,「メッセージフロー・グラフ」という形で表現可能であり，このグラフが，メッセージフロー解析時に利用される．

また，従来の並列オブジェクト指向モデルには,「メッセージを待つ」という概念は存在せず，さらには，受信したメッセージに対しては，必ず1つの処理の流れが生成されていた．これに対して，メッセージフロー・モデルはメッセージの待ち合わせ機構を持つ．このメッセージの待ち合わせ機構を装備することで，オブジェクトが処理を開始するために2つ以上のメッセージを必要とし，それらがオブジェクトに揃った時に処理を開始するというデータフロー・モデルに見られるような処理手段の表現やデータの待ち合わせを，そのデータを持つメッセージの待ち合わせとして表現することが可能となる．

3 MIMD 型並列計算機上へのモデルのマッピング

メッセージフロー・モデルの同期機構は，そのモデルに基づいて実現する MIMD 型並列計算機上での実行効率を高めるための1つの手段であった．しかしこのモデルにも，これを並列計算機上にマッピングする際，i) あるオブジェクトが割り当てられた PE への処理の集中; ii) PE 台数の非効率的な使用; といった問題が生じる可能性がある．

この問題はメッセージフロー・モデルが並列オブジェクト指向モデルから受け継いでいる性質に起因する．これに対する解決策として我々は，図2に示すように，従来1つの「実体」として扱っていたオブジェクトのインスタンスを，役割分担によっていくつかの「機能実体」に分ける方法を検討している．

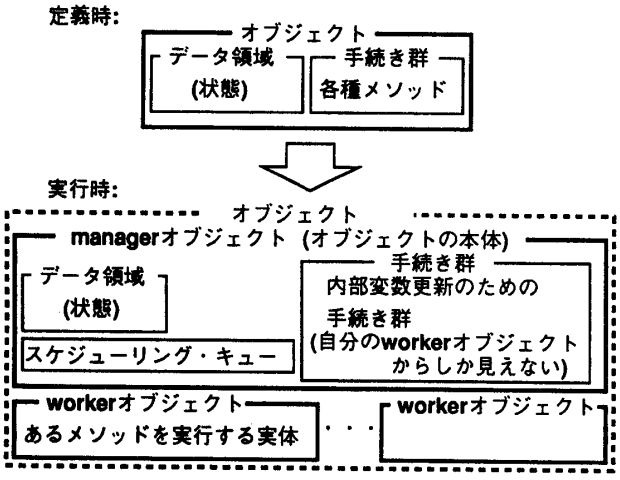

図 2: オブジェクトの機能実体への分割

この方法に基づく実行モデルにおいて，1つのオブジェクトは，1つの manager オブジェクト（ manager ）と複数の worker オブジェクト（ worker ）の組として実現される．そして，この manager と worker はそれぞれ，1つのインスタンス（実体）として存在する．

manager は，従来のオブジェクトと同様にその内部データを一括管理し，メッセージを受信する．しかし，この manager は，メッセージに対するメソッドの実行は行なわない．メソッドの実行を行なうのは，worker である．worker は，manager が受信メッセージに対するメソッドの実行を開始しようとする時点で生成され，そのメソッドの実行を担当する．そして，メソッドの実行が終了すると消滅する．このため，計算実行時のある時点において，1 つの manager のインスタンスに対する複数の worker のインスタンスが存在する可能性がある．

基本的に，これらの「機能実体」は，それぞれ別の PE に割り当てる．したがって，この方法は，我々が開発を進めている並列計算機システムの持つ PE 台数をより有効に活用し，負荷を自然に分散させることができる可能性がある．

なお，このような方法を従来の並列オブジェクト指向モデルの下で採用すると，本来そのモデルが想定していた処理とは異なった処理が生じることになる．つまり，同期機構の問題がより顕在化してくる．これに対して，メッセージフロー・モデルにおいては，この方法を採用したことによるメッセージフローの本質的な変化は存在せず，メッセージフロー・モデルが有する同期機構をそのままの形で利用できる．

また，メッセージフロー・モデルでは，従来オブジェクトの内部状態として存在していたようなデータの一部を，可能な限りメッセージによって「運ぶ」ことによって，このデータの排他制御に関する処理のボトルネックを緩和することもできる．

並列計算機上へのモデルのマッピングに関しては，他にも処理単位の大きさ（粒度）という検討項目がある．メッセージフロー・モデルでの粒度はプログラム中のオブジェクト定義におけるメソッドの記述に左右される．このメソッドが，手続き型プログラミング言語における手続きや関数といった概念と本質的に同種であるために，比較的大きな処理単位となる可能性もあり，また，並列処理記述を行なう面でもある種の妨げになると考えられる．そこで我々は，メソッド記述によってこれらの問題が生じる可能性を抑えるための手段として，関数呼び出しを意識したような返信待ちをともなうメッセージ通信手段を提供しないことにした．これによって，以下の効果が期待できる．

- メソッド記述におけるメッセージ送信部分の前後で，処理の意味的な違いが生じることが予想され，このことが，処理単位としてのメソッドを，手続きや関数よりも小さくする結果につながる．
- メッセージ・フローという概念がより強調できる．

4 おわりに

現在我々は，メッセージフロー・モデルを我々が開発を進めている MIMD 型並列計算機システムに適用するための種々の検討を行なっている．

参考文献

[1] 藤井，新實，柴山: 超並列計算機システムにおける要素プロセッサ・アーキテクチャの検討, 電子情報通信学会・技術研究報告, Vol.90,No.144,CPSY90-45,pp.43-48 (1990).