

共有メモリ型マルチプロセッサにおける ストリーム計算の実行形式の評価[†]

5D-8

青柳 洋一, 中林 嘉徳, 岩田 竜一, 上原 稔, 森 秀樹^{††}
東洋大学工学部情報工学科^{†††}

1 はじめに

共有メモリ型マルチプロセッサの普及しつつある現在、その有効利用を目指した並行オブジェクト指向言語の効率の良い実装への期待が高まっている。並行オブジェクト指向言語には、プロセスを用いてオブジェクトを実現したもの、スレッドを用いたもの [1]、仮想機械によるシミュレート [2] で実現したものなどがあるが、それぞれ一長一短があり、実行方式の違いによる比較評価は意義のあることである。

並行オブジェクト指向計算におけるオブジェクトの実現方法による比較を [3] では通信時間の評価によって行ったが、本研究ではさらに主処理時間の評価を加えてより正確さを高め、効率の良い実装方式選択のための一般則を導いた。また、オブジェクト間通信の手段としてのストリーム通信の有効性を確認した。以下、処理効率、実行方式のモデリング、実行方式の比較評価、まとめの順に述べてゆく。

2 処理効率について

効率が良いということについて、(a) 処理時間が短い、(b) 資源を有効に利用している、という2通りの解釈ができるが、本論文では前者を優先にして考えてゆく。

処理全体にかかる時間を以下のように表すことができると考えられる。

$$\text{全処理時間} = \text{主処理時間} + \text{通信時間} + \text{その他} \quad (1)$$

主処理時間: 計算に本質的に費やされる時間

通信時間: オブジェクト間の通信時間

その他: コンテキスト切替え等のオーバーヘッドや待ち時間等

また、処理時間に影響する要因としてスケジューリング¹が考えられる。

式 (1) の要素は以下の式で表せると考えられる。

$$\text{主処理時間} = L/f(N_{CPU}) \quad (2)$$

$$\text{通信時間} = \sum_{\text{通信方式}i} (C_i/g(N_{CPU})) * n_i \quad (3)$$

L : 主処理の負荷の大きさ

C_i : 通信方式 i の通信コスト

n_i : 通信方式 i の通信回数

f, g : プロセッサへの負荷分散効果やスケジューリングの影響を表す関数

N_{CPU} : プロセッサ数

- 主処理が大きなオブジェクトの場合はスケジューリングによる影響が大きく、式 (2) により効率の良い実行方式を選択することができ、
- 通信頻度の高いオブジェクトの場合は、通信時間による影響が大きく、式 (3) で効率の順序関係を求めることができると考えられる。

3 オブジェクトのモデリング

実行方式による比較を行うため、プロセス (以下 Proc または P)、スレッド (以下 Thr または T)、並行シミュレート (Simulated Object, 以下 SO または S) の3つの実行単位の組合せでモデル化を行った。各実行単位の特徴を表1に示す。なお、オブジェクト間の通信方式としてストリーム通信を用いるものとする。

3つの実行単位を組み合わせて表2の6モデルを構成し、比較を行った。

[†]Evaluation of Execution Method for Stream Computing on Shared Memory Multiprocessors

^{††}Yoichi Aoyagi, Yoshinori Nakabayashi, Ryuichi Iwata, Minoru Uehara, Hideki Mori

^{†††}Department of Information and Computer Sciences, Toyo University

¹プロセッサへの負荷分散の効果や処理効率に影響する。

実行単位	特徴	ストリーム構成
プロセス (Prc,P)	計算資源確保の単位 プロセス移送が可能	pipe
スレッド (Thr,T)	スレッド間資源共有	共有変数 lock
Simulated Object (SO,S)	上位の実行単位による 並行シミュレート 並列性なし	共有変数

表 1: 各実行単位の特徴

モデル	特徴
(1)Prc	Prc のみでオブジェクトを実現
(2)Thr	Thr " "
(3)SO	SO " "; 並列性なし
(4)P × T	Thr の集合を Prc でまとめたモデル
(5)P × S	(1) のオブジェクトを SO で置換
(6)T × S	(2) " "

※ ×印は左の実行単位による右側のグルーピングを示す。すなわち (グループ) × (構成要素)

表 2: モデル構成

4 評価

通信回数、主処理の負荷の大きさが既知の処理を表 2 の各モデルで実現し、パラメータとしてプロセッサ数 (1,2,4 個)、グループ数 (2~8)、モデル形態 (表 2 の 6 モデル) を変化させて処理時間を計測した。なお、評価は Omron Luna88k(1/2/4CPUs) 上で行った。

順位	1プロセッサ		2プロセッサ		4プロセッサ	
	予測	実測	予測	実測	予測	実測
1	SO	SO	Thr	Thr	Thr	Thr
2	T × S	P × S	P × T	P × T	P × T	P × T
3	P × S	T × S	T × S	P × S	Prc	Prc

表 3: 効率順序 (上位 3 位) の予測・実測比較 (3 グループ)

評価結果から以下のことが言える。

- 予測・実測比較において予測順位が実際と異なるところがあるが、僅差であり、計測誤差の範囲内と見られることもできる。

- 1プロセッサにおける、SO モデルの効率の良さを確認ができた。
- マルチプロセッサにおいては、オブジェクト数がある数 (今回の調査では 20 程度) を越えるまでは Thr、P × T モデルの効率が良く、それを越えると T × S、P × S のように、並行シミュレートを組み合わせたモデルが効率が良いという傾向が得られた。

5 まとめ

- 通信方式をストリーム通信とすることによってデータサイズ、オブジェクト間の同期を意識する必要がなくなりプログラミングが容易となった。また通信の処理への影響が明確となり、安定した処理効率を得易くなった。
- 我々が開発中の並行オブジェクト指向言語 NET/C は、ストリーム通信に基づき、実行時のモデルとして複数のバリエーションが選択可能である。本研究により、ストリーム通信の利点を確認し、効率の良い実行方式選択のための指針が得られた。
- 今後、さらに異なるアーキテクチャのマシンにおける評価を行ってゆく。また、back ground で行われている他の処理との相互の影響を調べ、より評価の一般性を高めてゆく。

参考文献

- [1] R. Trehan, N. Sawashima, A. Morishita, I. Tomoda, A. Inoue, and K. Maeda. Concurrent Object Oriented C (cooC). *ACM SIGPLAN NOTICES*, 1993. February.
- [2] Yasuhiko Yokote. *The Design and Implementation of ConcurrentSmalltalk*. World Scientific, 1990.
- [3] 青柳洋一, 中林嘉徳, 岩田竜一, 上原稔, 森秀樹. 共有メモリ型マルチプロセッサにおける並行オブジェクトの実行系の評価. 情報処理学会研究報告, 1994. 94-PRG-18.