

OCoreにおける4つ組+1構造による行列共同体の実装

2E-2

御手洗 潔 田中 二郎
筑波大学

1 はじめに

われわれは、簡素で拡張性の高い並列オブジェクト指向言語として、RWCP(新情報処理開発機構)で開発が進められているOCoreに注目している[2][4]。OCoreは従来の並列オブジェクト指向言語の基本部分に加えて、共同体、メタレベルアーキテクチャ、グローバルGCなどを導入している[4]。

OCoreの特徴的な概念の1つである共同体はOCoreのオブジェクトが失ったメッセージ処理の並列性を提供する。共同体はオブジェクトの集合をインデックスの任意のn個の組によって一意に決定可能なn次元空間($n \geq 1$)に配置し、管理するための枠組みである。さらに、共同体のインスタンスはそのメンバオブジェクト間でのバリア同期、メンバオブジェクトのスロットに対するリダクションなどの大域的操作が可能である。また、共同体と似た概念をもつ言語にCA(Concurrent Aggregates)[1]等がある。

OCoreによるプログラミングではアプリケーション内の定型的な並列性(データ並列)を共同体で、非定型的な並列性(コントロール並列)をオブジェクトにより記述することが可能である。

しかしながら、共同体のデータ並列性の記述能力に注目した場合、粒度の大きさがメンバオブジェクトの大きさによって決定されることになる。

したがって、内在する並列性をアプリケーション毎に的確に記述を行なうには共同体の個々のメンバオブジェクトを構造化する必要がある。共同体及びメンバオブジェクトはオブジェクト指向計算における、配列構造の1つの拡張であると考え、アプリケーションとして行列演算を適用することにした。

本研究では、行列演算の記述の容易性と実行の効率化をめざして、行列演算の並列実行のために4つ組+1構造による行列共同体の実装方法を示す。

2 行列共同体

まず、与えられた正方行列Aを行方向、列方向ともに等しく等間隔になるように区切る。そして、それぞれの部分行列(以後、これをブロック行列[3]と呼ぶ)を成分とするブロック化行列を考える。さらに、ブロック行列からの2次元共同体のメンバオブジェクトへのマッピングを行列共同体と定義する。

このようにして行列と共同体の対応を考えることにより、行列演算に内在しているデータ並列性を共同体を使って記述できる。

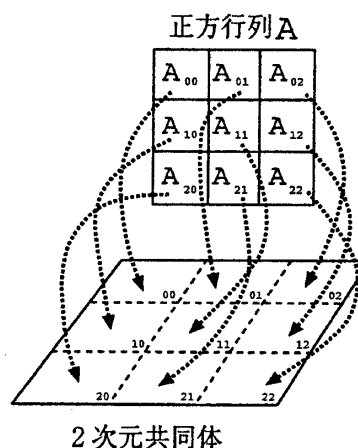


図1: 正方行列Aの2次元共同体へのマッピング

3 行列共同体の実装の概要

われわれは、行列(ベクトル)の基本演算等を並列に行なうために行列共同体を定義し、その実装のために4つ組+1構造を考案した。

3.1 4つ組+1構造

4つ組+1構造の共同体での外観は図2に示すように、共同体の各メンバオブジェクトにブロック行列と同等のサイズ $\times 4+1$ の領域を確保したものである。図3に共同体のメンバオブジェクトの(i,j)成分の4つ組+1構造を拡大したものを示し、行列の値がどのようにセットされるか説明する。

通常、行列共同体を生成し、データセットメソッドにより正方行列Aをセットしたとすると図3に示すように、4つ組の左上に正方行列Aの(i,j)ブロック行列成分がセットされる。 $w_0 \sim w_2$ は作業のための領域であり、tmpは行列の転置等の操作の際に使用する。

3.2 4つ組+1構造の操作

行列共同体のメンバオブジェクトの4つ組+1構造における操作は以下のように分類することができる。

An Implementation of the Matrix Community using Quartet+1 Structure in OCore
Kiyoshi MITARAI and Jiro TANAKA
University of Tsukuba
Tennoudai 1-1-1, Tsukuba-shi, Ibaraki-ken, 305 Japan

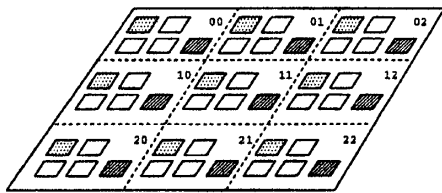


図 2: 行列共同体上の 4 つ組 +1 構造の外観

- 他の行列共同体のメンバオブジェクトの 4 つ組 +1 構造間での操作
- 自分自身の行列共同体の他のメンバオブジェクトの 4 つ組 +1 構造間での操作
- 任意のメンバオブジェクトの 4 つ組 +1 構造内での操作

それぞれデータのコピー、スワップ、和、差、積などの行列の基本演算を行なうことができる。

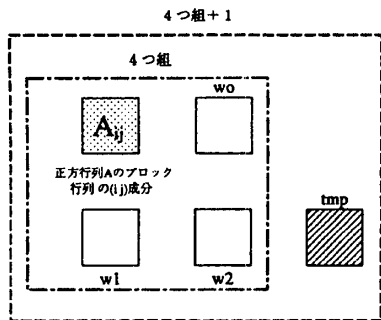


図 3: 行列共同体の (i, j) 成分の 4 つ組 +1 構造の詳細

3.3 行列共同体上の行列演算アルゴリズム

行列共同体上の 4 つ組 +1 構造を用いた各種行列演算を記述する際には、ブロードキャストにより全てのメンバオブジェクトのブロードキャストハンドラが同時に起動されることが基本となる。

したがって、各々のブロードキャストハンドラ内でのエラー処理などの重複した処理の一元化は、ブロードキャストハンドラをプログラム中の全ての場所から呼び出せるグローバル関数でラッピングする形式とすることにより解決した。このことにより、一つのブロードキャストハンドラだけでは記述できない処理の記述や API (アプリケーションプログラムインタフェース) に柔軟性を持たせることも可能になった。

次に、行列共同体上の 4 つ組 +1 構造を用いた行列の乗算アルゴリズムを説明する。行列共同体を用いた行列の乗算は行列共同体の列ベクトル単位の積の計算を列ベクトルの個数分行なうことにより実現している。図 4 に 3 × 3 の場合の行列共同体の最初の列ベクトル部分の積が求まる様子を示す。

ステップ 1 では行列共同体 B の最初の列ベクトルの値を行列共同体 A の各行ベクトルの w_0 にコピーす

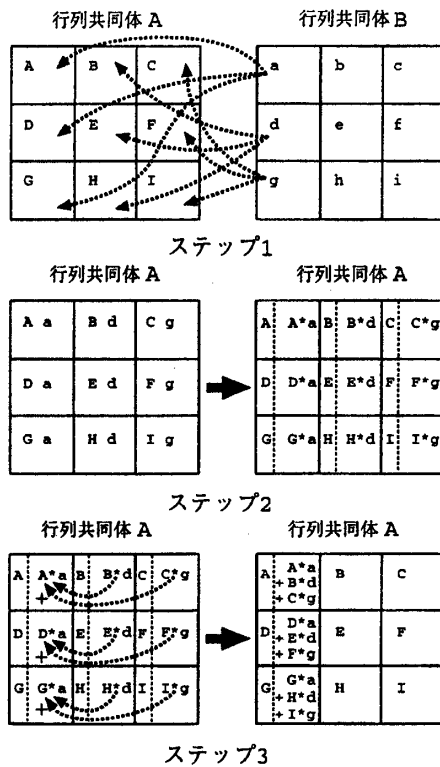


図 4: 行列共同体上の行列の乗算

る。ステップ 2 では行列共同体 A の全ての成分で $w_{1ij} = A_{ij} \times w_{0ij}$ を行なう。ステップ 3 では行列共同体 A の第 1 列ベクトルの w_{2i1} へ行方向に第 1 ~ 第 3 列ベクトルの w_{1ij} を足し込む操作を行なう。

4 おわりに

本研究では、並列オブジェクト指向言語 OCore における 4 つ組 +1 構造による行列共同体の実装方式について述べた。さらに、行列の乗算による行列共同体上の 4 つ組 +1 構造の適用例を示した。その結果図 4 のステップ 2 において行列演算の持つ並列性を引き出せたと考えている。今後は実アプリケーションへの適用による評価をふまえた行列共同体の改良を行なう予定である。

参考文献

- [1] Chien, A.A.: *Concurrent Aggregates*, The MIT Press, 1993.
- [2] Real World Computing Partnership: *OCore Reference Manual*, Feb., 1996.
- [3] 小国力 (編): 行列計算ソフトウェア - WS、スーパーコン、並列計算機 -, 丸善, 1991.
- [4] 小中裕喜, 友清孝, 前田宗則, 石川裕, 堀敦史, Jörg Nolte: 並列オブジェクト指向言語 OCore における共同体の拡張, In SWoPP'95, 1995.