

4 U - 4

漸進的並列処理のための 配列アクセス・パターン解析手法*

三田 勝史, 朝倉 宏一, 渡邊 豊英†

{sanda, asakura, watanabe}@watanabe.nuie.nagoya-u.ac.jp

名古屋大学大学院 工学研究科 情報工学専攻‡

1 はじめに

プログラムの並列化処理においてよく用いられる手法として、DO ループ並列化処理がある。これは、DO ループ中のイタレーション間から並列性を抽出し、イタレーションを並列実行することで処理時間の短縮を図る手法である [1]。しかし、イタレーション間のデータ依存関係が複雑な DO ループ文も多く存在する。それらの DO ループ文は DO ループ並列化手法が適用できず、逐次実行しなければならない。

イタレーション間の依存関係が複雑な DO ループ文に対する並列化手法として、我々は、イタレーション間の並列性ではなく、DO ループ文間の並列性を抽出することを考える。我々が並列化の対象とする数値計算プログラムでは、通常同一の配列データに対する DO ループ文は複数存在する。したがって、単一 DO ループ内のイタレーション間の並列性を抽出できなくても、同一配列データに対する参照・更新処理の局所性や規則性を抽出し、それによって DO ループ文を並列実行できれば、プログラム全体の処理を効率化することができる。我々は、上記の DO ループ文間の同一配列データに対する参照・更新処理の規則性を抽出し、並列実行させる漸進的並列処理を提案している [2]。本稿では、DO ループ文における配列データへの参照・更新処理の規則性を配列データに対するアクセス・パターンとして定義し、アクセス・パターンの表現方法、解析手法について述べる。

* Access Pattern Analysis for Array Data in Incremental Parallel Processing

† Katsushi SANDA, Koichi ASAKURA and Toyohide WATANABE

‡ Department of Information Engineering, Graduate School of Engineering, Nagoya University

2 配列アクセス・パターン

先にも述べたように、DO ループ文に対して漸進的並列処理を適用するためには、DO ループの配列データに対する参照・更新処理の規則性を配列アクセス・パターンとして抽出しなければならない。我々は、DO ループ文の実行とともに配列データ内の操作領域がどのように推移するかをアクセス・パターンとして定義する。そのため、DO ループ文実行中のある時点において配列データ内で既操作領域と未操作領域を分割する境界線と、DO ループ文の実行による境界線の移動方向を用いてアクセス・パターンを表現する (図 1)。

Two Dimensional Array Space

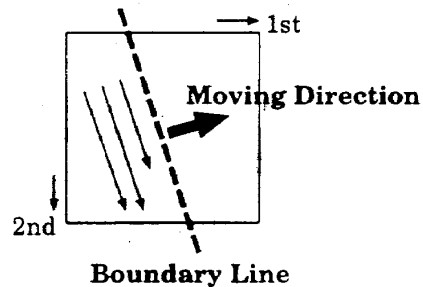


図 1: 配列アクセス・パターンの表現

3 アクセス・パターン解析手法

本節では、二重ループ中の二次元配列データに対する以下のような DO ループ文を対象として、アクセス・パターン解析手法について述べる。

- ループの下限が 0、増分が 1 に正規化されている。

- 配列インデックスがループ変数の線形結合で表されている。

対象とする DO ループ文の例を図 2 に示す。

```

for (i=0; i<I; i++) {
  for (j=0; j<J; j++) {
    .....
    A[2*i+1][i-j+1]=...;
    .....
  }
}

```

図 2: 対象とする DO ループ文の例

まず、DO ループ間で共有される配列がある場合、参照・更新操作される配列要素が共有されているか否かを判定する。個々の DO ループ文で参照・更新操作される配列要素の集合を配列インデックス空間上の領域として表現し、その領域を配列アクセス領域と呼ぶ。また、DO ループ間で共有されるアクセス領域を共有アクセス領域と呼ぶ。アクセス領域は、インデックス変化ベクトルと呼ばれる二本のベクトルで規定される平行四辺形で表現される。配列アクセス領域の模式図を図 3 に示す。実線

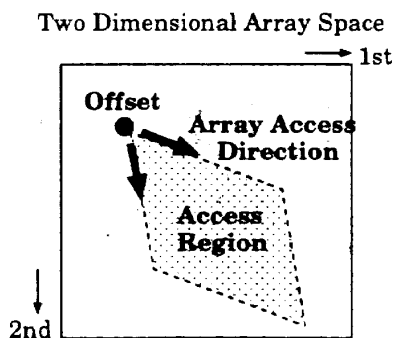


図 3: アクセス領域の表現法

の四角形は配列インデックス空間で、破線の平行四辺形は配列アクセス領域を示している。また、図中の矢印はインデックス変化ベクトルを、オフセットは配列インデックスの初期値を表し、平行四辺形の各辺の長さはループのイタレーション回数より求められる。インデックス変化ベクトルは、ループ変数をそれぞれ変化させたときの、配列インデックスの変化の方向をベクトルで表現したものである。つまり、個々の DO ループのイタレーションの実行によっ

て配列インデックスがどのように変化するかを表している。図 2 の場合、ループ i の配列アクセス方向は $(2, 1)$ 、ループ j のインデックス変化ベクトルは $(0, -1)$ となる。

プロセス間で共有アクセス領域を求めた後、アクセス・パターンが一致するか否かを判定する。アクセス・パターンは、既操作領域と未操作領域を分割する境界線とその移動方向で表現される。二重ループの内側のループ変数によるインデックス変化ベクトルを用いて境界線は表現される。また、境界線の移動方向は、二重ループの外側のループ変数のインデックス変化ベクトルの、境界線に対する直交成分で表現される。これを境界線移動ベクトルと呼ぶ。つまり、二重ループの外側ループのイタレーションが実行されるたびに、境界線が移動ベクトルの方向に推移する。

アクセス・パターンの一致度の検証においては、境界線移動ベクトルの傾きのみが必要で、その大きさは考慮する必要がない。したがって、DO ループ間でアクセス・パターンが一致しているか否かの検証には、大きさ 1 に正規化した境界線移動ベクトルの内積が利用される。内積が 1 に近いほど、アクセス・パターンは一致していると判断される。

4 おわりに

本稿では、配列データ分割による漸進的並列処理適用における配列アクセス・パターン解析について述べた。

今後の課題として、多次元の配列や、複雑な入れ子構造の DO ループにも対処できるようなアクセス・パターンの解析手法を検討する必要がある。

参考文献

- [1] Hans Zima and Barbara Chapman: “スーパーコンパイラ”, オーム社, 村岡洋一訳, P. 390 (1995).
- [2] 三田勝史, 朝倉宏一, 渡邊豊英: “配列データ分割による漸進的並列処理手法の適用”, 並列処理シンポジウム JSPP '98, p. 147 (1998).