

1 G-4

並列計算機用 F O R T R A N の
D O ループ実行制御方式齊藤鉄郎¹⁾ 迫田行介²⁾日立マイクロコンピュータエンジニアリング¹⁾ 日立製作所システム開発研究所²⁾1. はじめに

分散型共有メモリ方式並列計算機^{1), 2)} H 2 P 用の、D P 計算モデルに基づいた指示文方式並列化 F O R T R A N (ParaFORT) のトランスレータにおけるコピーの割付け方式とその制御および D O ループの実行制御方式について報告する。

D P 方式³⁾ では要素プロセッサ(以下 P E と略す)間に跨る参照はコピーを使用する。コピーの割付け方式には参照時のオフセットを一定にする方式と転送するオフセットを一定にする方式があるがここでは実行の高速化のために参照オフセットを一定化する方式を採用する。そのためにコピーのサイズ拡大という問題が発生するが、それを解決するコピーの割付け方式を報告する。

D P 方式では分散された配列はローカルなインデックスでアドレスする。ここではグローバルなインデックスで記述された D O ループをローカルインデックスによる制御に変換する手法を報告する。

2. コピーの問題

図 1 に 4 台の P E に配列 A (1 2) を分散割付けした状態を示す。各 P E には局所配列を 3 個ずつ割り付けることになる。ここで元のプログラムで A (J - 2) を参照している場合、各 P E には参照用のコピー配列 C A を割り付ける。コピーの割付けを機械的に実行すると図 1 のように局所配列をそのまま参照する P E に割り付けることになる。問題はコピー配列の参照オフセットが P E により変化することである。実行されるプログラムを図 2 のようなものとするとき A (J) に対して、P E 1 では C A (J - 1) を参照するのに対して、P E 4 では C A (J) を参照する。これはオリジナルの配列では局所的なインデックスは P E 4 から P E 1 へラップアラウンドするとき切り上がるのに対してコピー配列では P E 2 から P E 3 に移るとき切り上がるのが原因である。

2. 1 コピー制御

この問題を解決するためにコピーの転送先の P E 番

号がラップアラウンドすると転送のオフセットを切り上げる。コピー転送のオフセット値は主記憶中に取りられる F O T²⁾ (Fan Out Table) により制御される。F O T を作成するとき転送先 P E に従ってオフセットを切り上げる。

2. 2 コピーの割付け

コピーの転送をこのように変更した場合の問題は図 3 に示すようにコピー配列のサイズがオリジナルより大きくなることである。このためにコピー配列の定義は図 4 に示すように領域を確保してから E Q U 文で宣言する。

3. D O ループ変換

各 P E で D O ループを実行するときはループは局所的なインデックスで制御する。元のプログラムでは D O ループの初期値、終値、増分値はグローバルインデックスで表現されているのでこれを局所的な表現に変換する。ここではホスト側で一括して変換する方式と各 P E で個別に変換する方式を報告する。元のプログラムでの初期値、終値、増分値をそれぞれ e_1 , e_2 , e_3 とし、P E の総数を N (N は 2 のべき乗) とする。ここでは簡単の為、 e_3 と N は互いに素(簡単に言えば e_3 が奇数ということ)と仮定する。

3. 1 変換方式 1 (一括変換)

この方式ではサイズが N の配列 I E 1 と I E 2 を用意し、I E 1 (J) と I E 2 (J) に番号 J の P E の初期値と終値を格納する。なお、増分値は P E によらず一定である。

① $e_1 + j * e_3$ $j = 0, 1, 2, \dots, N-1$ を考えると e_3 と N が互いに素という仮定から、これらは N を法とする剰余類を尽くしている。

従って $j = 0$ から $N-1$ について $X = e_1 + e_3 * j$ の属する P E 番号 $\alpha(j)$ を求めて I E 1 ($\alpha(j)$) に $e_1 + j * e_3$ の値を格納する。最終的に I E 1 (J) に J 番目の P E での初期値が入る。ここで I E 1 (J) に格納した値はグローバルなインデックスの値

なので、局所的な値に変換しておく。

次に e_2 と e_1 の値からグローバルな最終インデックスの値 β を求める。初期値の場合と同様に

$$\textcircled{2} \quad \beta - j * e_3 \quad j = 0, 1, 2, \dots, N-1$$

を考えることにより終値を計算することができる。

3.2 変換方式2 (個別変換)

変換方式1ではすべてのPEでの初期値と終値を求めるのでPE側で計算する方法としては適当でない。変換方式2としてPE側で自分の初期値と終値のみを求める方式を与える。

まず e_3 と N が互いに素という条件から次の条件を満たす2つの正整数 K_1 と K_2 が存在する。

$$\textcircled{3} \quad K_1 * e_3 + K_2 * N = 1 + e_3 * N$$

今、 e_1 が割り付けられたPE番号を p_1 とし、このPEの番号を p とする。このとき

$$\textcircled{4} \quad e_1 + (N + p - p_1) * K_1 * e_3$$

番目の要素はこのPEに割り付けられる。それを確かめるには上の数値から1を引いてから N で割った余りが $p-1$ になれば良いがこれは $\textcircled{3}$ の関係から導かれる。

$\textcircled{4}$ の e_3 の係数

$$\textcircled{5} \quad (N + p - p_1) * K_1$$

は $\textcircled{1}$ の j の値としては大きすぎるがこれを N で割った余り

$$\textcircled{6} \quad \text{MOD}((N + p - p_1) * K_1, N)$$

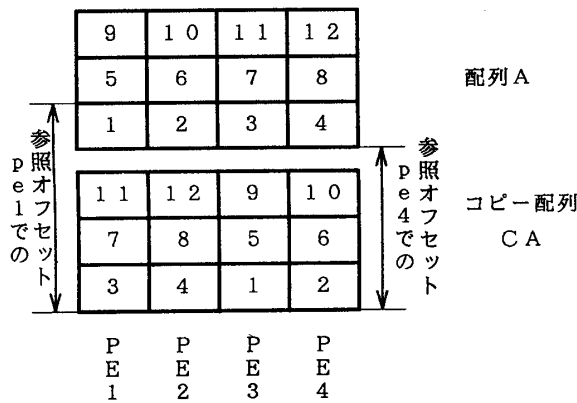


図1 転送オフセット一定の場合

```

DO 100 J=3, 12
  X=A(J-2)+...
  .....
100 CONTINUE
    
```

図2 プログラムの例

は $\textcircled{1}$ の j の中に含まれ、このPEで実行される最初の j の値である。

終値は e_2 とここで求めた先頭インデックスの値

$$\textcircled{7} \quad e_1 + \textcircled{6} * e_3$$

と増分値 $e_3 * N$ から計算できる。

4. おわりに

並列計算機H2Pにおけるコピーデータの割付け方式とコピーの更新方式について検討し、コピーの参照オフセットを一定にする方式を採用した。

各PEでのDOループ実行をローカルインデックスで制御する為にDOループの初期値、終値をグローバルな値からローカルな値に変換する方式について、ホスト側で一括して変換する方式とPE側で個別に変換する方式について報告した。

5. 参考文献

- 1) 村松、他：データパラレル計算モデルの提案
第36回情報処全大講演論文集 (I) 145-146、1988
- 2) 村松、他：分散型共有メモリをもつデータパラレル並列計算機のアーキテクチャ、第38回情報処全大講演論文集 (III) 1478-1479、1989
- 3) 斉藤、他：分散型共有メモリのためのFORTRANインタフェイス、第38回情報処全大講演論文集 (III) 1480-1481、1989

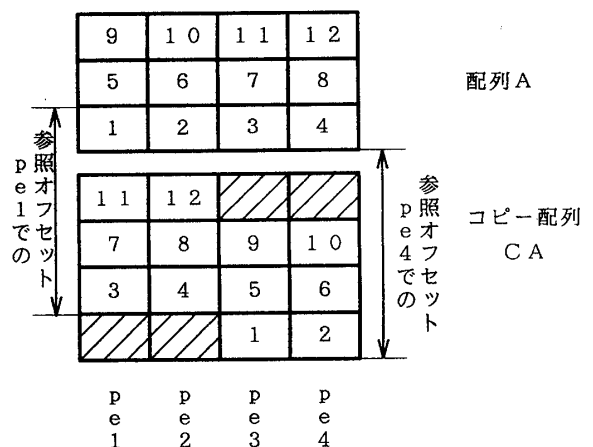


図3 参照オフセット一定の場合

```

DIMENSION A(3)
DIMENSION CA(3)
DIMENSION AREA(4)
EQUVALENCE (CA(1), AREA(1))
    
```

図4 コピーの割付けの例