6W-4

並列処理システム－晴－におけるカラー管理方式

石井吉彦、安江俊明、山名早人、村岡洋一
(早稲田大学 理工学部)

1．はじめに

本稿では、並列処理システム－晴－［1］におけるカラー管理方式の一提案を行なう。－晴－では、マクロブロック［1］という処理単位内で動的データ駆動方式を採用している。動的データ駆動方式ではループの処理にカラーを用いる。しかし、カラーは有限であるため、カラーの資源管理が必要となる。カラーの資源管理、即ち、カラーの回収・再割当に関して従来の方式では、「カラーのオーバフロー時に新しいループを生成する方法」が提案されている［2］。しかしながら、ループ生成のオーバヘッドが大きいという問題を持つ。また、計算機資源は有限であるから、計算機資源以上のカラーを用いても、処理速度向上は望めない。即ち、計算機資源に見合ったカラーを使用すれば良い。これらの点をふまえて、本稿では、必要以上のカラーを使用せず、カラーの回収・再割当のオーバヘッドを削減したループ処理方式を提案する。

以下では、まず、ループ本体に対しデータフロー解析を行ない、カラーの必要個数（Lで表わす）を求める。そして、カラーのオーバフローを回避し、Lで制限されたカラーを回収・再割当するループ処理方式を示す。なお、今回はLが計算機資源以下の場合について報告する。

2．ループ処理方式

ループ処理中に用いるトークンを定義し、－晴－におけるループ処理方式を詳述する。

2-1. トークンの定義

【ループ制御トークン】逐次型言語では、ループを制御するために、ループ制御変数を用いる。一方、データフローでは、ループ制御変数に加えて、ループの各インデックス間でのトークンを区別するためにカラーを用いる。本稿では、この様に、ループを区別するためにカラーを持ち、かつ、ループ制御変数をデータとして持つトークンを、他のトークンと区別して「ループ制御トークン」と呼ぶ。ループ制御トークンは、図2-1 に示す様な構成をとる。

【データトークン】DATA部にループ制御変数以外のデータを持つトークンを「データトークン」と呼ぶ。

2-2. ループ制御トークンの振舞

図2-2 のループ文に基づき、 ループ制御ﾄｰｸﾝの振舞を説明する。ループ処理方式を簡単に図2-3 に示す。（同図中の①～④は、本稿の番号と対応する。）

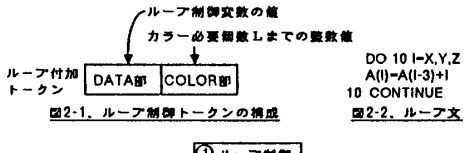

図2-1. ループ制御トークンの構成

```
DO 10 I=X,Y,Z
A(I)=A(I-3)+I
10 CONTINUE
```

図2-2. ループ文

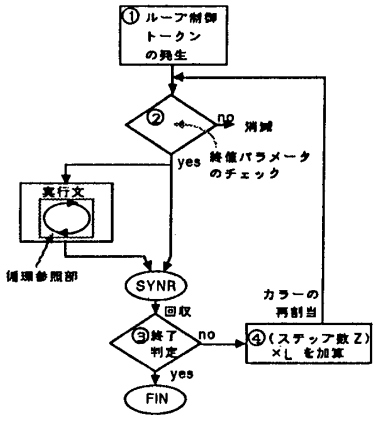

図2-3. ループ処理方式の説明図

〔例えば、図2-1 の場合で、カラー必要個数L＝3 として説明する。〕

①ループ制御トークンをL（＝3）個発生させる。ループ制御変数の初期値ﾊﾟﾗﾒｰﾀ Xからステップ数Zずつ増加させた3つ値(X, X+Z, X+Z×2)をそれぞれの ﾙｰﾌﾟ制御ﾄｰｸﾝのDATA部に格納し、カラーの1から3の値（1, 2, 3)をそれぞれの ﾙｰﾌﾟ制御ﾄｰｸﾝのCOLOR 部に格納する。（図2-4 参照）

② ﾙｰﾌﾟ制御ﾄｰｸﾝのDATA部が上限（終値ﾊﾟﾗﾒｰﾀ Y）を越えている場合、ﾙｰﾌﾟ制御ﾄｰｸﾝを消滅させる。③実行文の中でループ制御変数という ﾙｰﾌﾟ制御ﾄｰｸﾝの役割を終えると、その ﾙｰﾌﾟ制御ﾄｰｸﾝを回収する。回収の際にはループ回数をカウントし、ループの終了判定を行なう。

④回収された ﾙｰﾌﾟ制御ﾄｰｸﾝのDATA部に、（ステップ数Z）×（I (=3)）を加算する。加算したものを新しい ﾙｰﾌﾟ制御ﾄｰｸﾝとして再割り当てする。（図2-5 参照）

⑤処理②③④を繰り返す。

この様なループ制御をすることにより、第4節で述べる様な利点を得る事ができる。

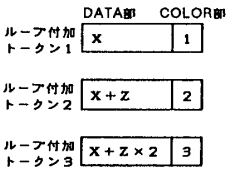

図2-4. ループ制御トークン発生

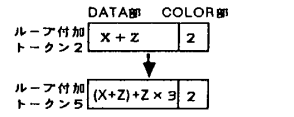

図2-5. ループ制御トークンの回収・再割当

3．ループ制御ﾄｰｸﾝの発生個数（カラー必要個数）Lの決定方法

データフローでは、ループ制御にカラーを用いるため、通常、ループ繰り返し回数分だけカラーが必要となる。しかし、全てのインデックスに対して、カラーを割り当てたとしても、それらインデックス間で並列に計算できなければ、カラーの意味はない。即ち、並列に実行できるインデックスの個数ほどのカラーがあれば十分である。従って、カラー必要個数Lの決定要因は、ループ本体の並列性を制限している「文間の依存距離」となる。

ループは大きく分けて DOALL型とDOACROSS型がある。 DOALL型の場合、ループ本体に依存距離がないので、カラー必要個数Lの決定要因は無い。DOACROSS型の場合、循環参照部を持つため、循環参照部の依存距離がカラー必要個数Lの決定要因となる。

以下では、まず、ループ本体をデータフロー解析するために必要な用語を定義する。その後、ループ本体の分類を行ない、カラー必要個数Lを求める。

3-1. 用語定義

【データフローグラフ上の循環参照】ループ本体を表わすデータフローグラフにおいて循環参照とは「ある1つの演算ノードに着目し、その演算ノードをAとした時、演算ノードAの出力であるデータトークンがいくつかの演算ノードを経て、再び、演算ノードAの入力となる事。」である。このデータトークンの流れる過程で、データトークンは1以上のカラーの増加が行なわれる。

【リング】循環参照を構成する閉路を「リング」と呼ぶ。

【リング依存距離】リングにおけるカラーの増加はそのリング上にある全ての依存距離の和と等しい。この和を「リング依存距離」と呼ぶ。（図3-1 参照）

【世代】1つのリングについて考えた時、そのリング内で並列実行可能なデータトークンの個数は、リング依存距離Rに等しい。この時、ある時刻において並列実行可能なデータトークンの集合を「世代」と呼ぶ。なお、世代は、時間軸方向に増大するものとする。（図3-2 参照）つまり、カラーをR増加させる事によって、データトークンの世代が1増加する。

【πブロック】ループ本体を表わすデータフローグラフにおいて、リングのみで構成される部分を「πブロック」［3］と呼ぶ。

A Method of Color Management
for Parallel Processing System -Harray-
Yoshihiko ISHII,Toshiaki YASUE,Hayato YAMANA
and Yoichi MURAOKA
WASEDA University

【πブロック分割アーク】πブロックとπブロックがリングに属さないアークのみでつながっている場合、このアークで依存性が断ち切られている。このアークを「πブロック分割アーク」と呼ぶ。(図3-3 参照)

3-2. ループ本体の分類とカラー必要個数Lの決定

ループ本体は、大別して、1つのπブロックで構成されるものと、複数のπブロックで構成されるものに分類される。さらに、それぞれの場合について2つに分類される。1つのπブロックは、1つのリングと複数のリングに分かれ、複数のπブロックは、πブロック分割アークの依存距離の有無により分類される。以下では、これら4つの場合に対して、カラー必要個数L.の決定方法を述べる。

Ⅰ. 1つのπブロック

Ⅰ-1. 1つのリング

結論【L＝リング依存距離Rの倍数】

リング依存距離をRとすると並列実行可能なデータトークンはR個となる。S世代目のデータトークンは、Sの値に関係なく常にR個あり、${}_sT_1, {}_sT_2, \cdots, {}_sT_r, \cdots, {}_sT_R$とする。ここで、${}_sT_r$ のカラーを$C({}_sT_r)$と記述する。

一世代目のデータトークンを区別するには、カラーが全て異なればよいので、カラーの個数はR個必要となる。また、カラーの歯抜けを起こさないようにするため、カラーを1から逐次データトークンに割り当てる。この時、

$$C({}_1T_r)=r \qquad (r=1,2,\cdots,R)$$

となる。同様に、S世代目のデータトークンを区別するには、カラーの個数はR個必要で、

$$C({}_sT_r)=C({}_sT_1)+r-1 \qquad (r=1,2,\cdots,R)$$

となる。

次に世代間の区別について考える。世代1の${}_1T_r$ と世代Sの${}_sT_r$ を区別するには、カラーは世代1から世代Sまでの間でS×R個使用すれば良いので、

$$C({}_sT_r)=C({}_1T_r)+(S-1)\times R$$

となる。世代1から世代Sまでの間で世代が区別されるので、世代S+1 では再び世代1のカラーを使用でき、

$$C({}_{s+1}T_r)=C({}_1T_r)$$

となる。つまり、S×R個のカラーにより、S世代にわたる世代の区別が可能であり、同様に、次のS世代にわたる世代の区別が可能である。以上の事は、明らかに任意の整数値Sで成り立つので、リングが1つの場合、カラーの必要個数LはRの倍数となる。

Ⅰ-2. 複数のリング

結論【L＝各リング依存距離Rの公倍数】

M個のリングがあり、それぞれのリング依存距離を$R_1, R_2, \cdots, R_M$ とするとⅠ-1. よりカラーの個数は$R_1, R_2, \cdots, R_M$ 全ての倍数でなければならないので、カラーの必要個数Lは$R_1, R_2, \cdots, R_M$ の公倍数となる。

Ⅱ. 複数のπブロック

Ⅱ-1. 「πブロック分割アークの依存距離」＝0

結論【L＝πブロックごとのLの公倍数】

πブロック分割アークの依存距離が0ならばπブロック間の世代は変わらないので、カラーの必要個数Lは、Ⅰ. より求めたπブロックごとのカラーの必要個数の公倍数となる。

Ⅱ-2. 「πブロック分割アークの依存距離」≠0

結論【L＝πブロックごとのLと付加リングの依存距離との公倍数】

πブロック分割アークをリングにすれば、1つのπブロックに変換できる。つまり、このπブロック分割アークと向きを異にし、同期ノードを伴ったアークを付加すればよい。πブロック分割アークと付加アークにより構成されるリングを「付加リング」と呼ぶ(図3-4 参照)。付加リングを付ける事によって、Ⅰ-2. に帰着するので、カラーの必要個数Lは、πブロックごとのカラーの必要個数と付加リングの依存距離Rとの公倍数となる。

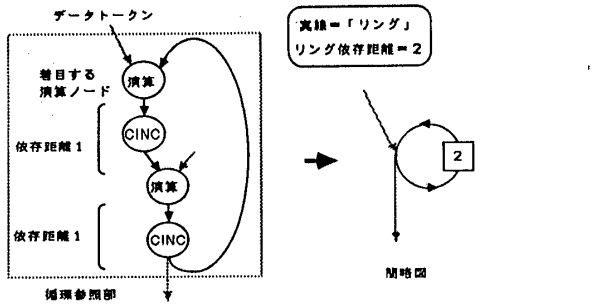

図3-1. リングの説明図

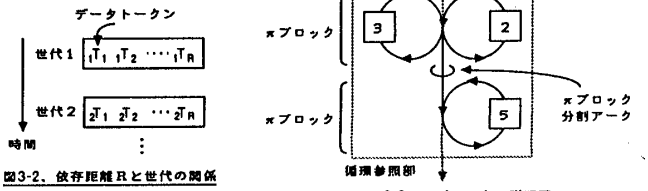

図3-2. 依存距離Rと世代の関係

図3-3. πブロックの説明図

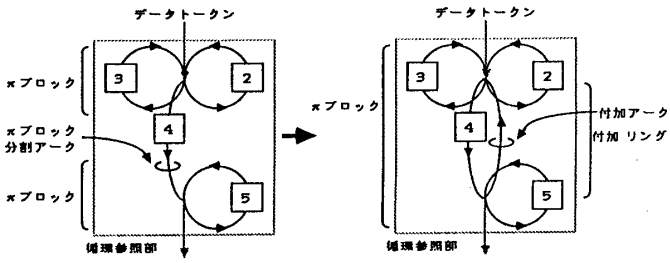

図3-4. 付加リングの説明図

4. 本方式の利点

ⓐ今すぐ実行されないと全体の実行時間に影響を与える命令を「クリティカルな命令」と呼ぶ[4]。データ駆動方式において、命令は非決定的に行なわれるため、有限な計算機資源を有効に利用するためには、クリティカルな命令をそうでない命令よりも優先しなければならない。従来の方式ではカラー発生の命令は、クリティカルな命令でなく、ループ本体を遅らせてしまうが、本方式ではループ本体の並列度に見合うカラーを使用するため、カラーの回収・再割当の命令は、常にクリティカルな命令となる。従って、処理速度の向上が望まれる。しかし、本方式では、ループ制御トークンの回収・再割当(図2-2.③④)はカラー個々に対して行なわれるため、回収・再割当のオーバヘッドは並列実行可能なループ本体に対し分散している。この時、分散したオーバヘッドが、ループ本体を遅らせる事も考慮しなければならない。現在、これらの事を考慮した評価を行なっている。

ⓑ一つのループ処理で使用するカラーを1～L.と制限して使用しているので、L.+1以上のカラーを多重ループ処理やｻﾌﾞﾙｰﾁﾝ処理などの他の用途として使用できる。

5. おわりに

カラーの必要個数Lで制限されたカラーを、1個ごとに回収・再割り当てする事により、計算機資源を考慮したループ処理を述べた。また、Lが計算機資源以下の場合、本方式に問題はないが、Lが計算機資源以上の場合、課題が残る。現在、この事を考慮した本方式の評価と、多重ループ処理方式の考察を行なっている。

参考文献

[1] H. Yamana, T. Marushima, T. Hagiwara, Y. Muraoka : " System Architecture of Parallel Processing -Harray- ", Proc. of 1988 Int. Conf. on Supercomputing, pp. 76-89 (1988)

[2] 島崎、平木：" 科学技術用データ駆動計算機 SIGMA-1のループカウンタのオーバヘッド処理"，情報処理学会第30回全国大会論文集，pp. 255-256 (1985)

[3] U. Banerjee, S. Chen, D. J. Kuck, R. A. Towl : " Time and Prarallel Processor Bounds for Fortran-like Loops ", IEEE Trans. on Computers C-28, 9, pp. 660-670 (1979)

[4] 砂原、所：" データ駆動計算機の実行制御機構に関する考察"，ﾃﾞｰﾀﾌﾛｰﾜｰｸｼｮｯﾌﾟ, May, 1986, pp. 83-89