

# ループ並列化教育支援のための可視化

1N-6

山田光正 山下義行

筑波大学†

## 1 はじめに

並列コンピューティングの課題の一つに、逐次プログラムの並列化が挙げられる。これは、過去蓄積されてきた逐次型プログラムの資産を並列機上で活かすために欠かせない技術である。しかし逐次プログラムの並列化は、制御間やデータ間の依存性を充分考慮して負荷分散を行わなければならないため、簡単な作業とは言えない。

そこで我々は、一般に並列化による性能向上が大きいループ処理を対象に、並列化を支援する可視化システムを作成している。作成にあたっては、並列化プログラミングに不慣れな者でも簡単に扱うことが可能で、初心者がループ並列化を学ぶためにも使用できることを目標とした。このシステムは、2次元の簡単なウェイブフロント型ループをモデルに、ループ並列化に必要な依存情報などを画面に表示する機能、マウス操作でループの個々のイテレーションを各プロセッサに配置できるなどの機能を持つ。

本稿では、本システムの構成およびこれらの機能について概要を述べる。

## 2 システムの構成および概要

システムの構成を図1に示す。

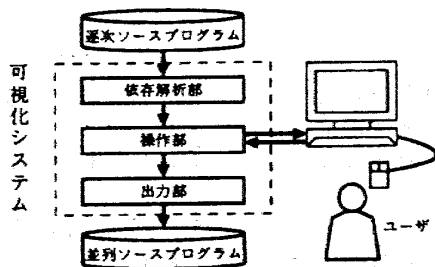


図1: システム構成

**依存解析部** ... 逐次ループ処理を記述したC言語プログラムを入力とし、イテレーション間のデータ依存解析を行なう。

**操作部** ... イテレーション間のデータ依存関係およびプロセッサ毎のイテレーションの実行順序を表示する。また、マウスによるイテレーションの実行順序および実行プロセッサの変更操作を受け付ける。

†An educational visualization for loop parallelization  
Mitsumasa Yamada, Yoshiyuki YAMASHITA  
(Univ. of Tsukuba)

出力部 ... 配置パターンおよびデータ依存関係をもとに、並列化したソースプログラム (MPI [1] 通信関数を含んだ SPMD 形式のプログラム) を作成・出力する。

次に、操作部の機能について詳しく述べる。

## 3 操作部機能

```
for(i=1; i<=4; i++) {
    for(j=1; j<=3; j++) {
        a[i][j]=a[i-1][j]+a[i][j-1]
            +a[i+1][j]+a[i][j+1];
    }
}
```

図2: サンプルプログラム

以下では、図2のプログラムを例に、操作部の各機能について説明する。

### 3.1 表示機能

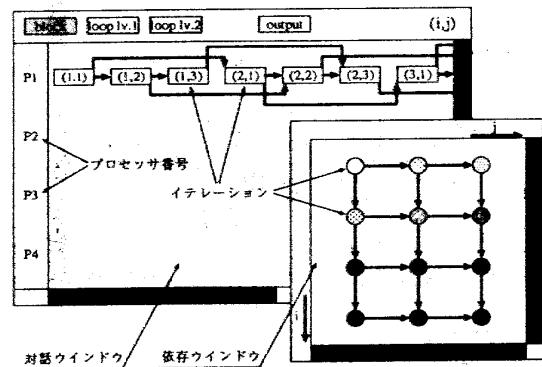


図3: 表示画面イメージ

図2のプログラムがシステムに入力された場合、操作部は図3に示すような画面を表示する。この表示画面は、二つのウインドウにより構成される。

依存ウインドウは、イテレーション間のデータ依存関係を表す。各小円はループ中の個々のイテレーションを表し、小円を結ぶ矢印は矢印先のイテレーションが矢印元のイテレーションに依存することを表している。(システムはイテレーション間で相互の依存を持つループを受け付けない。そのため、矢印は常に単方向である。) 各小円の色は実行順序を表しており、明→暗の順に実行されることを示している。

対話ウインドウでは、各プロセッサの処理するイテレーションを時系列で表している。各矩形は、ループインデックス  $(i,j)$  が矩形内の値を持つ時のイテレーションを表す。矩形間の矢印は、依存ウインドウの小円間の矢印と同じく、イテレーション間のデータ依存関係を表す。矩形の表示位置は、縦方向が実行プロセッサを、横方向が時間を表している。

図3では、全12個のイテレーションが単一のプロセッサ (P1) 上で逐次実行されることを示している。

### 3.2 対話操作機能

ユーザは対話ウインドウ上の各矩形をマウスで操作することで、イテレーションの実行プロセッサおよび実行順序を変更することができる。これは、操作したい矩形をクリックし、次に矩形の配置先をクリックすることで行なう。配置先が適正であれば(依存関係に反しなければ)依存関係に基づいて矩形の再配置/画面の再描画を行ない、不正であれば操作を受け付けない。

例として、図3の状態から  $(i,j)=(2,1)$  のイテレーションの実行プロセッサを P2 に変更する操作を、図4に示す。このとき表示画面は図4の (a) → (b) → (c) のように遷移する。画面 (b) における網掛け表示は操作する矩形の配置不可能な領域を示し、ユーザの操作を支援するものである。

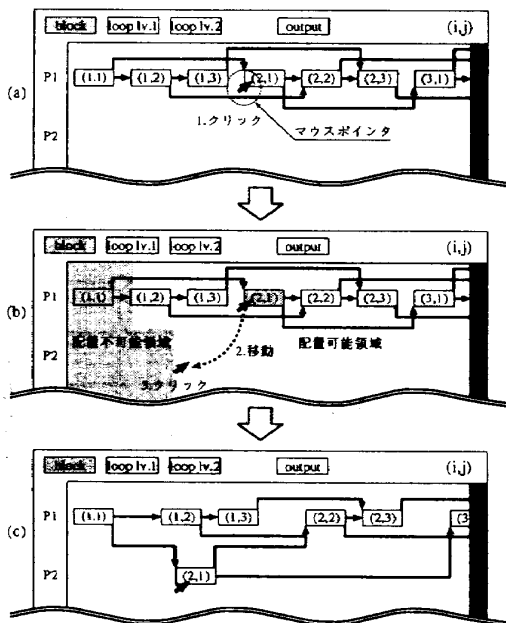


図4: 配置変更のマウス操作および画面遷移

更にユーザは、「block」 / 「loop lv.1」 / 「loop lv.2」のボタンを押すことで操作単位を変更することができる。例えば、図3において「loop lv.1」ボタンを押すと、内側ループ (jループ) で括られる矩形全てが操作単位になる。このとき、表示画面は図5のようになり、一度に操作できる矩形が破線枠内の3つになる。この機能は、イテレーション数が多数の際、個々の矩形を操作することによる

作業量の増加を軽減する。

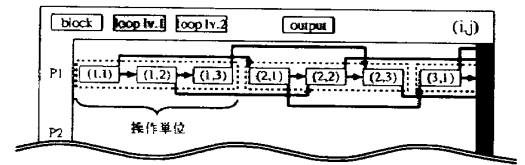


図5: 操作単位の変更

## 4 操作例

図3の状態から各矩形を操作し、最適なイテレーション配置に至った場合を図6に示す。

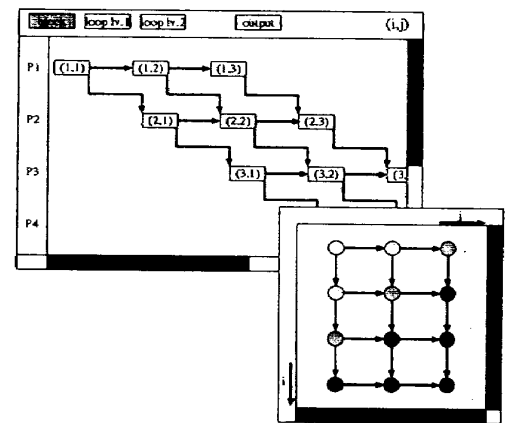


図6: 最適配置時の画面イメージ

図6において、依存ウインドウの各小円間のデータ依存パターンおよび対話ウインドウの各イテレーションの配置パターンを対比することにより、ウェイフロント型ループの並列化にはスキューイングが有効であることがわかる。

このように、ユーザは依存パターンと配置パターンを対比しながら対話的に並列化を進めることで、ループ並列化に対する体験的知識を得ることが可能である。

## 5 おわりに

本稿では、ループ並列化支援のための可視化システムについて、システムの構成と機能の概要を述べた。

本可視化システムはユーザに情報を提供するだけでなく、並列化を行なうための対話操作が可能である。そのため、前節でも述べたようにループ並列化に対する体験的知識を得ることが可能であり、初心者がループ並列化を学ぶ上で有効である。

## 参考文献

- [1] MPIフォーラム: MPI: メッセージ通信インターフェース標準 第1.1版 (1996)