

並列プログラミング用事例ベースの作成と 類似事例の検索・修正法

3L-4

松田 浩一 山崎 勝弘
立命館大学理工学部

1 はじめに

過去の並列プログラムを再利用することにより、並列プログラムの負担を軽減するためのシステムについて述べる。事例検索の鍵となる、並列アルゴリズムの特徴を抽出するための分類法、問題解析法、及び検索事例の修正法について説明する。

2 システム構成

図1にシステム構成を示す。ユーザはインデックスの各項目に沿って問題解析を行う。データ依存性、終了条件、トポロジから並列アルゴリズムを決定する。並列アルゴリズムは一般的に図2に示す4つに分類できる。各アルゴリズムに対する並列構造がシステムから提示され、インタラクション、並列化手法を基に、1つの並列構造を決定する。各並列構造には、その実行構造がBACS表現で与えられている。システムはインデックスを用いて最も類似した事例を事例ベースから検索する。スケルトンには、タスク分割、同期、相互排除、並列化手法など並列プログラムの最も重要な部分が骨格として含まれる。このスケルトンに計算部分などの肉付け/修正を施し、目的の並列プログラムを完成させる。完全にマッチする事例が検索できない場合でも、検索条件を緩めて、とにかく関連した事例を検索する。

インデックスは応用、仕様の他、次の8項目で構成する。

- アルゴリズム プロセッサファーム、分割統治法、プロセスネットワーク、繰り返し変換。
- 終了条件
- データ構造 ソースデータ、結果データ。
- タスク分割 分割対象、分割単位、分散方法。
- トポロジ ワーカー、トリー、パイプ、メッシュ...
- インタラクション signal, wait, バリヤ, mutex
- 並列化手法 パラレルリージョン, パラレルセクション, スレッドライブラリ。
- 実行構造 BACS 実行構造

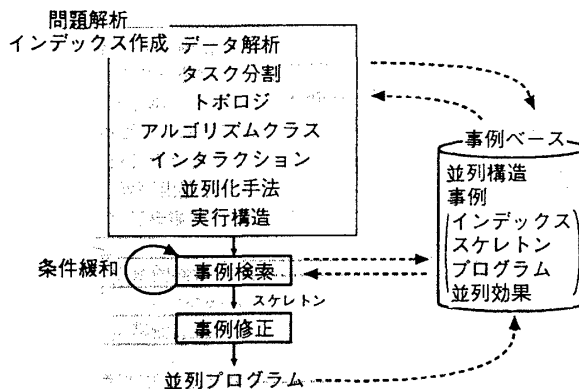


図1: 並列プログラミング支援システム

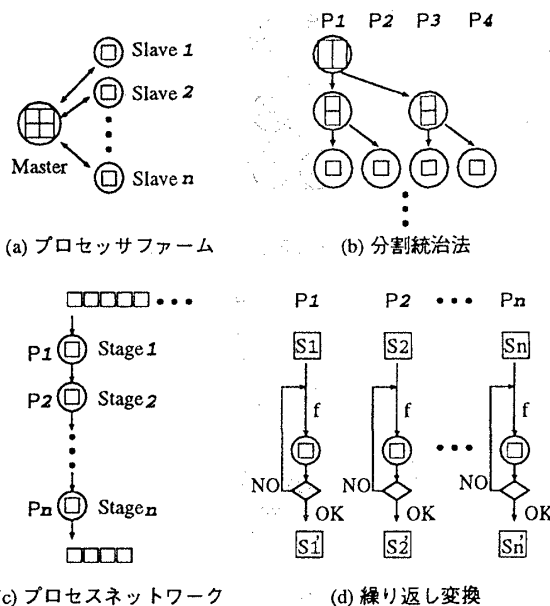


図2: 並列アルゴリズムの一般的分類

3 問題解析

3.1 タスク分割

タスクの分割要素となるものを発見するため、計算対象を分割し、単独に計算できる最小の計算単位(単位要素)を明らかにする。単位要素の計算を行うために必要とされるデータ(ソースデータ)と、その計算結果(結果データ)の格納方法を明らかにした後、それらのデータ構造を決定する。

A Case Base for Parallel Programming and Retrieving,
Adapting Method of Relevant Cases
Kouichi Matsuda and Katsuhiro Yamazaki
Ritsumeikan Univ.

単位要素をプロセスに割り振ることにより、タスク分割が実現できる。よって、計算対象、ソースデータ、結果データが分割の対象候補となる。該当するものに対して、(分割対象、分割単位、分散方法)を示す。

分割、分散方法には、全体のコピーを全プロセスに割り振る(共有)、ブロックに分割したものを割り振る(ブロック)、一定の周期に沿って順番に割り振る(サイクリック)、およびブロックに分割したものをサイクリックに割り振る(ブロック&サイクリック)の4つの方法が存在する。

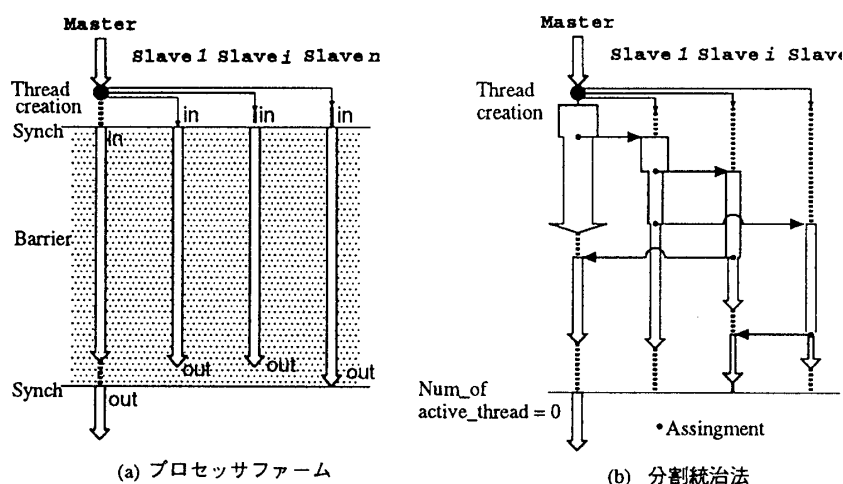


図3：並列構造

3.2 並列アルゴリズムの選定

各プロセスは、ソースデータを他のプロセスから受け取り、それに対する結果データを別のプロセスへと送る。このようなデータの流れから生じるプロセストポロジを決定する。トポロジの形態とプロセス間のデータ依存性から、問題に対する並列アルゴリズムを次のような条件で決定する。

プロセッサファーム ソースデータは他のプロセス上のソースデータ、結果データと依存関係、包含関係がなく独立である。

分割統治法 ソースデータは親子関係にある親プロセスのソースデータと包含関係にある。

プロセスネットワーク 結果データは1列に並ぶプロセスの方向に流れ、パイプライン処理される。

繰り返し変換 結果データは別のプロセス上でソースデータとして参照される場合がある。

3.3 並列構造の選定

並列アルゴリズムに対する並列構造がシステムにより提示される。これにより、スレッドの動き、データの流れ、同期の取り方などを視覚的に捕らえ、アルゴリズムの実行構造の理解を助ける。図3に、(a)バリエーション同期を用いたプロセッサファーム、(b)分割統治法の並列構造を例として示す。図2のアルゴリズムクラスに対する並列構造の中から、最も問題に適する並列構造を選択する。それらの実行構造はBACS表現で与えられている。並列アルゴリズムは、デーモンコール、インタラクション、計算の3つのイベントで構成される。BACS表現では、これらイ

ベントの実行順序と、各イベントに参加するプロセスを書き並べ、それらを簡単な略語と記号を用いて実行構造を表現することができる。[2]

インタラクションは次の3パターンに分類される。

1. 2プロセス間のインタラクション。(signal,wait)
2. 3個以上のプロセス間の互いに相手を認識する直接的なインタラクション。(バリエーション同期)
3. 3個以上のプロセス間の間接的なインタラクション。(mutex)

4 事例修正

インデックスの選択項目を基に、システムは類似事例を検索し、そのスケルトンを提示する。スケルトンは複数のプロセスを並列に動かすための骨格部分であるため、単位要素の計算式、データ、終了条件などの計算部分をスケルトンが提示する枠組の中に組み込む。データの初期化、結果の出力など並列ブロック以外の部分は、逐次プログラムの場合と同じ形のまま組み込むことができる。

5 おわりに

現在、仮想共有メモリ並列マシンKSR-1上で、並列プログラミング用事例ベースの作成を行っている。事例の自動修正、並列構造の提示方法などが課題である。

参考文献

- [1] 古川, 山崎: “類似事例を用いた並列プログラミング支援”, 情報処理学会第50回全国大会 2J-3, 1994.
- [2] Burkhart H. et al.: “BACS: Basel Algorithm Classification Scheme”, Technical Report 93-3, Universität Basel, 1993.