4AH－3

事例ベース並列プログラミングにおけるインデックスの作成法

安藤 彰一　山崎 勝弘
立命館大学理工学部

1　はじめに

新たな並列プログラムを作成する際に，過去に作成した並列プログラムの中から最も類似した事例を検索し，それを再利用することによって並列プログラミングの負担を軽減させるシステムにおけるインデックスの作成方法について述べる．

2　システム構成

図１にシステム構成を示す．このシステムは，過去に作成した並列プログラムを事例として事例ベースに蓄積してある．各事例には，問題の定義，インデックス，スケレトン，プログラム，並列効果の各項目が含まれる．インデックスはプログラムの特徴を示し，データ構造，タスク分割，アルゴリズム，並列化手法，インタラクション，実行構造など１０項目が含まれる．スケレトンには並列プログラムの骨格，つまりその並列プログラム中で使われるデータ構造，タスク分割，同期，相互排除，スレッドなど一連の手順が記載されている．新たな問題に対する並列プログラムを作成する際に，ユーザはインデックスの各項目に沿って解析を行いながら，それらを作成する．システムはそのインデックスを用いて事例ベースの中から最も類似した事例を検索する．検索された事例のスケレトンにその問題に必要な変数や計算部分などの肉付け／修正を行う．

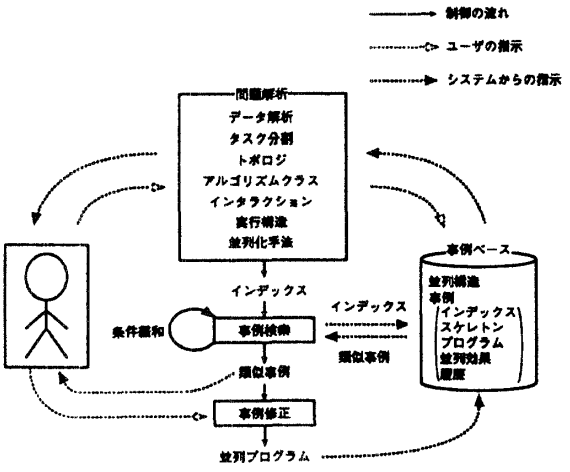

図１　並列プログラミング支援システム

Indexing in a Case-Based Parallel Programming System
Shoichi Ando and Katsuhiro Yamazaki
Faculty of Science and Engineering, Ritsumeikan University

問題に合わない部分は手動で部分的に修正して，並列プログラムを完成させる．このようにしてできた並列プログラムもまた，事例として事例ベースに蓄積される．

3　インデックスの作成法

3.1　インデックスの記述

インデックスの作成は各項目について順番に記載する．各項目とも図や説明を入れ，なるべく，ユーザが適当だと思うものを選択するだけでインデックスを記述できるようにし，簡単にインデックスが作成できることを目指す．また，例の中に当てはまるものがなかった場合はユーザが考えて記述する．

3.2　各インデックスの入力

インデックスの各項目は次に示すように順番に選択または入力していく．

1. 応用
   数値処理，ソーティング，経路探索，グラフ問題，画像処理，文字列照合，その他の７つから選択する．
2. 仕様
   文章または式で入力する．
3. データ構造
   計算の基になるソースデータと計算後の結果データを以下の手順で選択する．
   (a) 配列 or その他
   (b) １次元 or ２次元 or ３次元
   (c) 整数型 or 実数型 or 文字型 or 構造体
4. タスク分割
   タスク分割の入力ではソースデータ，結果データともに分割対象，分割単位，分散方法を入力する．分割対象はユーザが独自に入力する．分割単位，分散方法は以下の選択肢からそれぞれ選択する．

分割単位		分割方法
１要素	複数要素	ブロック
１行	複数行	サイクリック
１列	複数例	コピー
なし		

5. 終了条件
   ループの繰り返しが可変回のときに，その終了条件を指定する．固定回のときはなしとする．

6. トポロジ
ワーカー, ツリー, パイプ, メッシュ, リング, マスター&ワーカー, ハイパーキューブ, その他の8つから選択する。また, 図と説明を画面上に表示する.

7. 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, 繰り返し変換の4つから選択する. また, 図と説明を画面上に表示する（図2）.

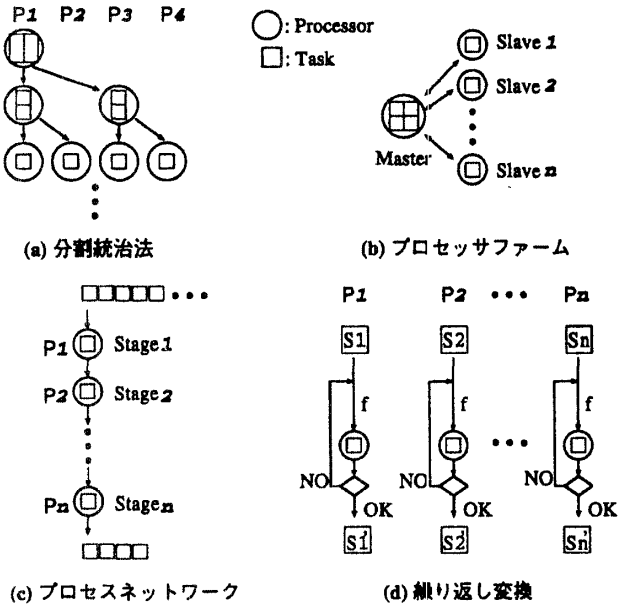

図2　並列アルゴリズムの分類

8. 実行構造
選択したアルゴリズムに対する代表的な実行構造の図とBACS表現を画面上に表示する. その中から最も適切な構造を選択する. プロセスネットワークの時は, その他にFSFE(First Start First End)とLSFE(Last Start First End)のどちらかを選択する（図3）.

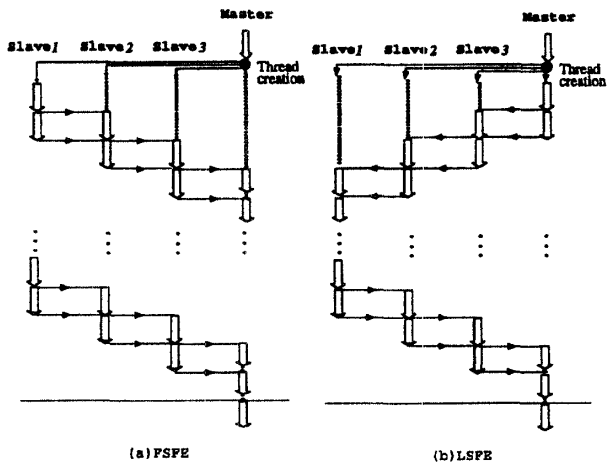

図3　並列実行構造

9. 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の3つから選択する. 実行構造が決まっているので, これらは自動的に決定される.

10. インタラクション
シグナル・ウエイト, バリヤ, mutex, なしの4つから選択する. 実行構造が決まっているので, これらは自動的に決定される.

4　インデックスの作成例と考察

4.1　インデックスの作成例

既に作成した「バブルソート」の並列プログラムのインデックスを作成してみた（図4）.

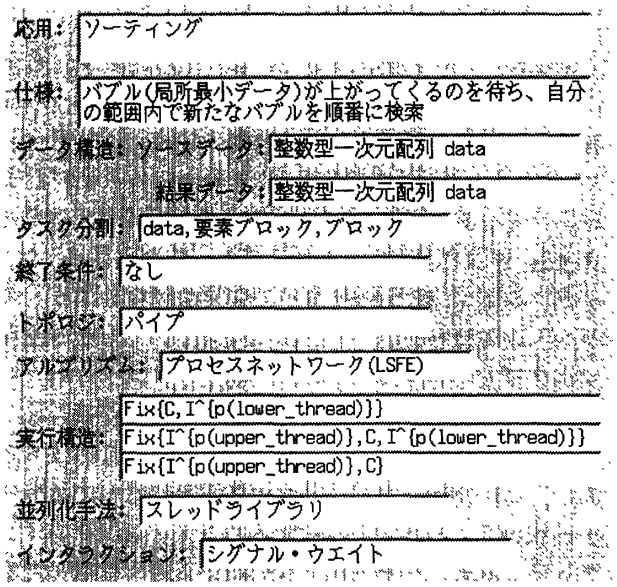

図4　バブルソートのインデックス

4.2　考察

入力の順番はタスク分割, 並列アルゴリズム, インタラクションとユーザが並列化する上で考えやすい順番にした. 各項目とも選択するだけで良いので簡単にインデックスを作成できた. アルゴリズムと実行構造の選択によって並列化手法とインタラクションが自動的に決まり, スケレトンの構造に最も影響を与えるので, これら2つの選択が非常に重要である. 一方, トポロジやアルゴリズムは図や説明文を示しているが, 初心ユーザは選択が困難かもしれず, 対策が必要であろう.

5　おわりに

現在, インデックスの入力部の作成を行っている. 今後, 事例の検索部と修正部の検討及び作成が必要である.

参考文献

[1] 山崎 勝弘, 安藤 彰一, 朝倉 啓之: "事例ベース並列プログラミングシステム", 並列処理シンポジウム JSPP'97, pp.117-124, 1997.