事例ベース並列プログラミングの評価

4N−6

米田 健治, 徳山 美香, 山崎 勝弘
立命館大学大学院理工学研究科

1 はじめに

新たな並列プログラミング時に, 事例ベースから最も類似した並列プログラムを検索し, それに手動で修正を加えて, 新たな並列プログラムを作成する方法について述べる. 本稿では, ビジネル暗号, ランレングス圧縮, 画像データの蓄積法, BM 法, ロンバーグ積分法の 5 つの問題について事例の再利用度を評価する.

2 システム構成

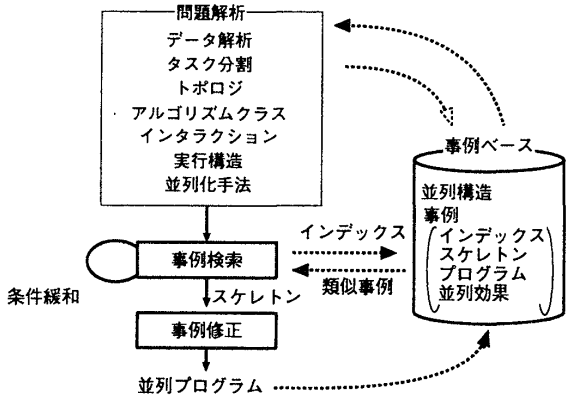

図 1: 事例ベース並列プログラミングシステム

ユーザはデータ構造, タスク分割, アルゴリズムなどの観点から問題解析を行い, インデックスを作成する. システムはインデックスを用い, 最も類似した事例を事例ベースから検索する. スケレトンには, タスク分割, 同期, 相互排除など並列プログラムの重要な部分が含まれ, これを修正し, 目的の並列プログラムを完成させる.

3 事例ベース並列プログラミング

3.1 ビジネル暗号

- 問題定義
  暗号の鍵を繰り返し用い, 鍵と平文の英字の番号の和を暗号文の英字の番号とする暗号化手法.
- 並列化の解析
  暗号化する TEXT の各文字ごとに, 鍵の対応する文字で独立に処理するので, プロセッサファームを用いる. インデックスを表 1に示す. 類似事例として KMP 法を用いた.

表 1: ビジネル暗号のインデックス

応用	暗号化
仕様	鍵と TEXT の英字の番号の和で暗号化する.
データ構造	ソースデータ: 文字型 1 次元配列 TEXT,Key
	結果データ : 文字型 1 次元配列 TEXT
タスク分割	ソースデータ: TEXT, ブロック, ブロック
	結果データ : TEXT, ブロック, ブロック
トポロジ	マスター&ワーカー
アルゴリズム	プロセッサファーム
実行構造	C^t
並列化手法	パラレルリージョン

3.2 ランレングス圧縮

- 問題定義
  AABBBBCCC → A2B4C3 のように, 文字とその文字の連続する数との組合せに置き換える圧縮方法.
- 並列化の解析
  TEXT が分割対象で, 暗号化と同様にプロセッサファームを用いる. 並列化手法がほぼ暗号化と同じなので, インデックスも暗号化とほぼ同じである.

3.3 画像データの蓄積法

- 問題定義
  1 画面に多くの点があるとき , それらの座標と付属情報を記憶する方法. 1 画面を 4 分割し, 各部分画面内の点の数が許容値になるまで, 再帰的に 4 分割を繰り返し, 各領域ごとにまとめて点を管理する.
- 並列化の解析
  再帰的アルゴリズムなので, 分割統治法を用いる. 画面情報を格納する 2 次元配列と許容値がソースデータとなり, 画面をブロックで等分割する. また, タスク分割のために相互排除が必要になるので, ミューテックスを用いる. インデックスを表 2 に示す. 類似事例としてクイックソートを用いた.

表 2: 画像データの蓄積法のインデックス

応用	画像データの蓄積法
仕様	領域を 4 分割し, 各領域の点をまとめて記憶. 各領域の点の数が, 許容範囲の数になるまで繰り返す.
データ構造	ソースデータ: 整数 2 次元配列 data, 整数型 pmax
	結果データ : 整数型 pos, count
タスク分割	ソースデータ: data, ブロック, ブロック
終了条件	分割された領域の点の数<= pmax
トポロジ	トリー
アルゴリズム	分割統治法
実行構造	$Cond\{I^{P(i)}_{waitNewTa}, Cond(C^{P(i)}, I^{P(i)}_{searchIdleTh})\}$
並列化手法	パラレルリージョン
インタラクション	ミューテックス

Evaluation of Case-based Parallel Programming
Kenji Yoneda, Mika Tokuyama and Katsuhiro Yamazaki
Ritsumeikan University

3.4 BM 法

- 問題定義

  文字列 TEXT 内で, PAT の位置を検出する. 各文字の出現位置と PAT の構造情報を用いる.
- 並列化の解析

  TEXT をブロックに分割する. 隣接ブロックの境界に PAT が存在する場合があるので, 境界付近を重複して割り振る. PAT 情報を全スレッドにコピーする. 各スレッドは独立に計算できるので, プロセッサファームを用いる. インデックスを表 3に示す. 類似事例として KMP 法を用いた.

表 3: BM 法のインデックス

応用	文字列照合
仕様	各文字の出現位置と PAT の構造情報を用いて, TEXT から PAT の位置を検出する.
データ構造	ソースデータ: 文字型 1 次元配列 TEXT 文字型 2 次元配列 PAT 結果データ : 2 次元配列 RESULT
タスク分割	ソースデータ: TEXT, 重複ブロック, ブロック PAT , 共有 結果データ : RESULT, 列, サイクリック
トポロジ	マスタ&ワーカー
アルゴリズム	プロセッサファーム
実行構造	C^t

3.5 ロンバーグ積分法

- 問題定義

  分割数 $2^i(i=0,1,...,k)$ の台形公式の積分近似値 $S_0^0, S_1^0, ..., S_k^0$ を初期値とする. $j=1$ として, $S_n^j = \frac{2^{2j}S_{n+1}^{j-1} - S_n^{j-1}}{2^{2j}-1}$ の S_n^j を $n=0,1,..,k-j$ に対して計算する. 同様に $j=2,3,4,...$ と変化させて, $|S_0^j - S_0^{j-1}| < eps$ になるまで計算を繰り返す.
- 並列化の解析

  タスク分割のソースデータとなる $S_0^j, S_1^j, ..., S_k^j$ を共有変数とする. S_n^j を S_{n+1}^{j-1} と S_n^{j-1} から求めるとき, j の増加と共に n が減少するので, S_n^j の計算を各スレッドにサイクリックに割り当てる. また, 各スレッドの計算が終了するまで待つために, バリア同期を用いる. 全スレッドがバリアにチェックインし, S_n^jの計算が終了し, 全スレッドがバリアからチェックアウトすると終了条件の判定に入る. アルゴリズムは繰り返し変換である. 類似事例として DKA 法を用いた.

3.6 並列効果

5 つの問題の並列効果を図 2に示す. それぞれの計測は, 暗号が TEXT40 万文字, 鍵 ABC, 圧縮が TEXT40 万文字, 蓄積法が 2000x2000 の画面で点 1 万個, 許容値 5, BM 法が TEXT10 万文字, PAT 数 40 個, 積分が式 $exp(0.5(1-x)(1-x))$, 区間 [0,1000000000], $k=15, eps=1.0e-8$ という条件で行った. プロセッサファームを用いた暗号, 圧縮, BM 法ではスレッド数に比例した速度向上が得られた. 一方, 蓄積法, ロンバーグ積分では 4 スレッド以上で速度向上が飽和している. 分割統治法と繰り返し変換では, 同期のためのオーバーヘッドが大きいと考えられる.

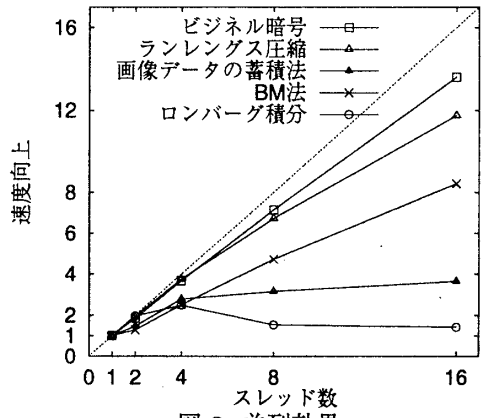

図 2: 並列効果

4 事例の再利用度

事例から作成した並列プログラムをスレッド, 同期, タスク分割, 単位計算について, 再利用, 修正, 新規作成した行数を表 4に示す. スレッド生成は約 95 %の再利用で再利用度が高い. 同期はプロセッサファームの場合, 修正の必要がなく, 分割統治法, 繰り返し変換の場合も事例と似通っていれば, 同様である. タスク分割は問題ごとに多様なので, 再利用度が低く, 多くの修正や新規作成が必要である. 単位計算はほぼ逐次プログラムを再利用できるが, タスク分割に関連して新規作成が必要になることがある.

表 4: 事例の再利用度 (行数)

		スケレトン			逐次
		スレッド	同期	タスク分割	プログラム
暗	再利用	8	1	4	33
	修正	0	0	5	0
号	新規	0	0	2	1
圧	再利用	8	1	5	28
	修正	0	0	4	0
縮	新規	0	0	0	14
蓄	再利用	30	9	3	22
	修正	0	0	18	0
積	新規	0	0	0	0
B	再利用	6	1	0	60
M	修正	0	0	0	0
法	新規	1	0	6	0
積	再利用	7	8	0	20
	修正	0	0	0	0
分	新規	2	0	1	0

5 おわりに

現在, 事例ベースを KSR-1 から分散共有メモリ並列マシン Exemplar 上に移植している. 再利用度の低かったタスク分割について再検討することが課題である.

参考文献

[1] 山崎, 安藤 : 類似事例を用いた並列プログラミング-LU 分解からナップサック問題へ, bit, Vol.30, No.7,pp.23-30,1998.

[2] K.Yamazaki and S.Ando:A Case-Based Parallel Programming System,International Symposium on Software Engineering for Parallel and Distributed Systems(PDSE '98), pp.238-245,1998.

[3] 山崎 他 : 事例ベース並列プログラミングシステム, 1997 年並列処理シンポジウム (JSPP'97) 論文集, pp.117-124, 1997.