

記号処理プログラムのベクトル処理法と その論理型言語プログラムへの適用

金田 泰

要 旨

Cray-1 を祖とするベクトル計算機は、従来、ほとんど数値計算専用につかわれていた。しかし著者は、HITAC S-810 などの第 2 世代ベクトル計算機がもっているおおくの機能は記号処理いいかえれば非数値処理に使用できることを確信した。それにもとづいて研究・開発された各種の記号処理のベクトル計算機における実行のための基礎技術について、この論文ではのべる。すなわち、第 1 に N クウィーン問題をはじめとするさまざまな解探索問題あるいは AI プログラムのベクトル処理を可能にする並列バックトラック法についてのべる。第 2 に、記号処理に特徴的なリスト、木、グラフなどの可変長データ構造を処理する複雑なくりかえし制御構造を、くりかえし構造の交換やくりかえし構造の 1 重化などにもとづいてベクトル処理可能にするベクトル化法についてのべるとともに、リストのベクトル処理に必要な基本演算のベクトル処理法についてのべる。第 3 に、グラフや共有要素をもつリストや木などの共有部分があるデータの処理や、複数データのハッシュ表への登録のように同一データへの同時かきこみをおこなう可能性がある処理をベクトル処理するための方法についてのべる。

また、これらの基礎技術の応用として開発されたベクトル化法とそれによる実測結果とをしめす。すなわちまず、Prolog で代表される逐次論理型言語によって記述された解探索プログラムのためのベクトル化法についてのべる。このベクトル化法の適用によって、S-810 による N クウィーン問題のプログラムの実行において逐次実行の 9 倍程度の高速化を実現した。また、逐次論理型言語および GHC で代表される並列論理型言語による素数生成問題のプログラムのためのベクトル化法についてのべる。このベクトル化法の適用により、逐次実行の 3 ~ 4 倍の高速化を実現した。また、入出力データのモードが確定した解探索などの逐次論理型言語プログラムから高速なベクトル処理をおこなうプログラムを生成することができる自動ベクトル・コンパイラのプロトタイプを開発したが、その構造と機能とについてのべる。これらの論理型言語プログラムのベクトル化法は、たかい加速率をえようとすれば限定された範囲のプログラムしか変換できないが、今後の開発によってたかい加速率とひろい応用範囲とを両立させられるようになることが期待される。

さらに、これらのベクトル処理法において使用されているマルチ・ベクトルというデータ構造についてのべる。リストなどのポインタを使用したデータ構造をマルチ・ベクトルに変換することによって、広範な記号処理をベクトル処理可能にすることができる。

これらの研究成果により、ベクトル計算機の一部の記号処理への適用可能性を実証し、ベクトル計算機の応用範囲の拡大を実現することができたとかんがえる。また、これら

の技術は、パイプライン型ベクトル計算機にとどまらず、今後飛躍的に発展することが予想される Connection Machine CM-2 のような SIMD 型並列計算機に応用することができるのかんがえられる。

謝 辞

この研究に協力していただいた以下の方々に感謝する。

東京大学の 和田 英一 教授には、指導教官として、この論文を洗練するのをてつたっていた。

現日立製作所中央研究所第 7 部の 菅谷 正弘 氏には、著者とおなじ第 8 部に所属していたとき共同研究者としてこの研究に参加していただき、とくに論理型言語プロトタイプ処理系の開発に従事していただいた。小島 啓二 氏には、筆者とおなじ中央研究所第 8 部に所属し、この研究にも関連がふかい HITAC M-680H IDP などの研究に従事している立場から、さまざまな示唆をいただいた。また、*N* クウィーン問題解探索の M-680H IDP 用のプログラム開発にあたって著者と共同作業していただいた。現慶応大学環境情報学部の 安村 通晃 氏には、日立製作所中央研究所在勤時にこの研究に関してさまざまな示唆をいただくとともに、筆者がカーネギー・メロン大学に客員研究員として在勤時に、中央研究所における非数値ベクトル処理研究のテーマ・リーダーを著者からひきついで研究をあらたな方向に発展させていただいた。Martin Nilsson 氏には、東京大学博士課程在学時にこの研究と関連のふかい FLENG Prolog の研究に従事し、それをつうじてさまざまな示唆をあたえていただいた。

また、現中央研究所副所長の 高橋 栄 氏には、この研究の開始時に中央研究所におけるユニット・リーダーとしてこの研究を支援していただくとともに、その後ソフトウェア工場から支援していただいた。現日立製作所システム開発研究所第 3 部長の 吉住 誠一 氏には、中央研究所における研究テーマとしてこの研究が発足して以来、筆者が所属する第 8 部 692 ユニットのユニット・リーダーとして、強力にバックアップしていただいた。さらに、現日立製作所常務の 武田 康嗣 氏、現中央研究所所長の 堀越 彌 氏、現中央研究所企画室の 千葉 常世 氏ほかの中央研究所の方々には、この研究を管理者の立場から支援していただいた。

最後に、妻 麗 はこの論文をまとめるにあたってさまざまな面でバックアップしてくれた。

目次

第1章 序論	1
1.1 研究の背景と動機	1
1.1.1 記号処理の高速化への要求のたかまり	1
1.1.2 ベクトル計算機とその機能の汎用機化	2
1.1.3 “汎用” 計算機の専用計算機に対する優位性	4
1.1.4 並列計算機とくらべての実用化時期のはやさ	5
1.1.5 SIMD 型並列計算機への応用可能性	6
1.1.6 プログラム変換による並列化の可能性	6
1.2 研究の目的	8
1.2.1 ベクトル計算機の応用範囲拡大	8
1.2.2 記号処理・論理型言語実行のベクトル計算機による高速化	8
1.3 論文の構成	10
第2章 解探索のベクトル処理	13
2.1 はじめに	14
2.2 逐次バックトラック計算法	16
2.3 AND ベクトル計算法 (AND 並列計算法)	18
2.4 OR ベクトル計算法 (OR 並列計算法)	20
2.4.1 完全 OR ベクトル計算法	20
2.4.2 並列バックトラック計算法	23
2.5 N クウィーン解探索の実行結果	27
2.6 並列バックトラック計算法におけるベクトル長増大法	34
2.7 まとめ	35
2.8 付録1: 測定用プログラムとその説明	36
2.9 付録2: N クウィーン解探索の測定データ	41
第3章 制御構造変換にもとづくベクトル化 — 1 リスト処理	45
3.1 はじめに	46
3.2 ベクトル化の基礎	49
3.3 ベクトル・リスト処理の戦略	51
3.3.1 プログラムの処理手順	51
3.3.2 くりかえし構造の交換	52

3.3.3	くりかえし構造の1重化	54
3.3.4	可変長リストのあつかい	54
3.4	リスト処理基本演算のベクトル処理法	60
3.4.1	データ型判定	60
3.4.2	リストの分解	62
3.4.3	リストの合成	63
3.5	エイト・クウィーンの Prolog プログラムへの適用	65
3.5.1	くりかえし構造の交換	65
3.5.2	くりかえし構造の1重化	68
3.6	評価	74
3.7	まとめ	76
第4章 制御構造変換にもとづくベクトル化 — 2 くりかえし構造交換法の比較評価		77
4.1	はじめに	78
4.2	くりかえし構造交換が必要な可変長くりかえしの例	80
4.3	可変長くりかえし構造交換のための技法	82
4.3.1	オーバラン無効化と終了判定コード生成	82
4.3.2	残存要素検出法	83
4.3.3	最大回数反復法	85
4.4	配列線形検索への適用例	87
4.4.1	スカラ処理アルゴリズム	87
4.4.2	ループ非交換版のベクトル処理アルゴリズム	88
4.4.3	マスク演算方式最大回数反復法によるベクトル処理アルゴリズム	89
4.4.4	インデクス方式残存要素検出法によるベクトル処理アルゴリズム	90
4.5	配列線形検索における性能実測結果とその検討	91
4.5.1	測定内容および条件	91
4.5.2	ループ交換にもとづくベクトル化の実用性	97
4.5.3	原外くりかえし回数への実行時間の依存性	97
4.5.4	最大回数反復法と残存要素検出法との比較	98
4.5.5	結果のまとめ	99
4.6	関連研究	101
4.7	まとめ	102
4.8	付録1: Fortran による配列線形検索のベクトル処理プログラム	103
4.8.1	スカラ処理版	103
4.8.2	ループ非交換版	103

4.8.3	マスク演算方式最大回数反復法版	104
4.8.4	インデクス方式残存要素検出法版	104
4.9	付録2: 配列線形検索の実測データ	106
4.10	付録3: 最大回数反復法におけるくりかえし回数 M_{max} の推定	108
第5章	共有部分がある複数データのベクトル処理法	111
5.1	はじめに	112
5.2	共有部分があるデータのベクトル処理の問題点	113
5.3	解決策としての上書きラベル・フィルタ法	117
5.3.1	上書きラベル・フィルタ法の原理	117
5.3.2	単一データかきかえの上書きラベル・フィルタ法	120
5.3.3	複数データかきかえの上書きラベル・フィルタ法	128
5.4	複ハッシングへの応用	132
5.4.1	複ハッシングのアルゴリズム	132
5.4.2	実行結果	137
5.5	番地計算ソート等への応用	139
5.5.1	番地計算ソートのアルゴリズム	139
5.5.2	実行結果	144
5.6	2分探索木への登録への適用	145
5.6.1	2分探索木への登録のアルゴリズム	145
5.6.2	実行結果	145
5.7	マーキング・PV 操作との類似点	146
5.8	関連研究	148
5.9	まとめ	149
5.10	付録1: Fortran による複ハッシングのプログラム	150
5.11	付録2: Fortran による番地計算ソートのプログラム	155
5.12	付録3: Fortran による2分探索木への登録のプログラム	157
第6章	論理プログラムへの応用 — 1 OR 並列性をふくむプログラムのベクトル化	159
6.1	はじめに	160
6.2	OR ベクトル計算法にもとづくベクトル処理法	162
6.3	決定的な手続きのベクトル化法	167
6.3.1	決定的な手続きの定義	167
6.3.2	手続きよびだしのベクトル化	167
6.3.3	手続き定義のベクトル化	170

6.4	非決定的な手続きのベクトル化法 (完全 OR ベクトル化)	179
6.4.1	手続きよびだしのベクトル化	179
6.4.2	非再帰的手続き定義のベクトル化	181
6.4.3	再帰的手続き定義の最適化されたベクトル化	186
6.5	並列バックトラックの実現 (並列バックトラック化)	191
6.6	評価	193
6.6.1	手動ベクトル化による評価	193
6.6.2	自動ベクトル化処理系の試作による評価	195
6.7	引数モード不確定のばあいへの適用拡大	197
6.8	関連研究	199
6.9	まとめ	202
6.10	付録: OR ベクトル化法の拡張と別方法	203
6.10.1	非決定的な手続きにおける解の蓄積法	203
6.10.2	カットのベクトル処理法	204
6.10.3	変数をふくむデータの複写	204
6.10.4	論理型言語による OR ベクトル処理プログラムの表現	205
6.10.5	OR ベクトル化アルゴリズム	211
6.10.6	関数型言語による OR ベクトル処理プログラムの表現	215
第 7 章 論理プログラムへの応用 — 2 AND 並列性をふくむプログラムのベクトル化		
		219
7.1	はじめに	220
7.2	データ構造変換にもとづくリストのベクトル処理法	222
7.3	Prolog による素数生成プログラムのベクトル処理	224
7.4	GHC による素数生成プログラムのベクトル処理	230
7.5	実行結果	235
7.5.1	測定に使用したプログラムについて	235
7.5.2	全体性能	235
7.5.3	部分ベクトル併合の効果	236
7.6	関連研究との比較	238
7.7	まとめ	240
第 8 章 論理プログラムの自動ベクトル化処理系とその中間語		
		241
8.1	はじめに	242
8.1.1	ベクトル化法の分類	242

8.1.2	処理系の構造	243
8.2	ベクトル中間語	245
8.2.1	ベクトル中間語の選択：論理型言語か関数型言語か	245
8.2.2	ベクトル中間語の機能	247
8.3	ベクトル化の方法	252
8.3.1	ベクトル化の例	252
8.3.2	ベクトル化の手順	256
8.4	実行方式	262
8.4.1	2つの実行方式	262
8.4.2	実行方式における3つの問題	263
8.5	ANDベクトル化の自動化について	265
8.6	まとめ	266
第9章	マルチ・ベクトル — ベクトル記号処理のためのデータ構造	267
9.1	はじめに	268
9.2	マルチ・ベクトル	269
9.3	マルチ・ベクトルの応用	270
9.3.1	解探索における使用	270
9.3.2	ストリーム処理における使用	271
9.3.3	ソートにおける使用	272
9.4	マルチ・ベクトルの機能	274
9.4.1	ベクトル再生成オーバヘッドの回避	274
9.4.2	バックトラック・並列処理の処理単位	274
9.4.3	ベクトル・パイプラインの有効活用	276
9.5	マルチ・ベクトルの操作法	278
9.5.1	部分ベクトルの分割	278
9.5.2	部分ベクトルの併合	279
9.6	まとめ	281
第10章	結論	283
10.1	研究成果	283
10.1.1	研究成果の概要	283
10.1.2	ベクトル計算機の記号処理への適用可能性の実証	283
10.1.3	複雑なくりかえし構造の変換によるベクトル処理法	285
10.1.4	可変データ構造の変換にもとづくベクトル処理法	286

10.1.5	共有部分があるデータのベクトル処理法	287
10.1.6	論理型言語プログラムのベクトル化法	288
10.2	結論	290
10.3	今後の課題	291
10.3.1	複雑なくりかえし構造の変換によるベクトル処理法	291
10.3.2	可変データ構造の変換にもとづくベクトル処理法	291
10.3.3	共有部分があるデータのベクトル処理法	291
10.3.4	論理型言語プログラムのベクトル化法	292
10.3.5	他の課題とまとめ	293
参考文献		295