

Title	Parallel TRAMのごみ集めの並列化に関する研究
Author(s)	斉藤, 嗣治
Citation	
Issue Date	1998-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1170
Rights	
Description	Supervisor:二木 厚吉, 情報科学研究科, 修士

Parallel TRAM のごみ集めの並列化に関する研究

齊藤 嗣治

北陸先端科学技術大学院大学 情報科学研究科

1998年2月13日

キーワード: 並列項書換え, 抽象機械, TRAM, マルチプロセッサ, 並列ごみ集め.

1 本研究の目的・背景

項書換えシステムは, 代数仕様言語, 関数型言語, 等式論理の証明などに幅広く応用できる計算モデルである. 等式を左辺から右辺への書換え規則とみなすことによって, 元来計算するという意味を持たないはずの等式を計算に用いるという考え方が基本になっている. このような考え方によって論理の世界と計算の世界を結びつけることができ, プログラムの検証や変換というような論理と計算の両方を用いる問題に対し有効に働くことができるモデルとなっている. また, 実装という観点から見ても計算機との相性は非常によく, 比較的容易に行うことができる. しかしながら, このような利点とは対照的に実際に実行するとその効率はあまりよいものではない. そのため, 実用にするためには数々の工夫が必要となる. 本稿では, 簡略化戦略に E-戦略を採用し, パターンマッチ用に弁別ネットを用いるといった特徴をもつ抽象機械 TRAM をさらに並列簡約が出来るように拡張し共有メモリ型マルチプロセッサに実装した Parallel TRAM を扱う. また Parallel TRAM の大きなボトルネックとしてごみ集めがあげられている. これは, Parallel TRAM がごみ集め時にグローバルな同期を必要とするということとごみ集めでは唯一つのプロセッサして働かないということが大きな要因となっている. そこで, 本稿では並列ごみ集めの考えを採り入れ, Parallel TRAM に実装することを試みる.

2 TRAM と Parallel TRAM

項書換え抽象機械 TRAM は, 特に実行速度の向上に重きをおいて設計・実装された抽象機械である. その特徴として, パタンマッチ処理部に弁別ネットを用いたパターンマッ

チの高速化や、その内部では項をすべて抽象的な機械命令列で表現しインタプリタで実行することができるといった事その他、E-戦略の採用によるユーザによる書換え順序の制御といったことがあげられる。

入力された書換え規則は、書換え規則のコンパイラにより規則の左辺を弁別ネットに、右辺を右辺のマッチングプログラムの雛型と戦略リストの雛型にコンパイルされ、それぞれ DNET および CR に納められる。弁別ネットはシンボルをキーとして分岐した木構造の事で、これを用いることによりマッチする規則を効率よく探すことが可能になる。入力項はマッチングプログラムと戦略リストにそれぞれコンパイルされ、CODE および SL に格納される。

Parallel TRAM では、TRAM が簡約の順番をユーザが決めることができるという E-戦略を用いているのを継ぎ、Parallel TRAM では E-戦略に引数項の並列簡約を明示的に指定できるように拡張した並列 E-戦略を用いている。

Parallel TRAM では、CPU、メモリといった資源を処理ユニットと呼ばれる単位に分けて管理する。TRAM と同じ構造をそのうちのひとつのユニットに割り当て、その他の処理ユニットに抽象命令を解釈・実行するインタプリタと簡約の際、内容が書換えられるおそれのある CODE、CL、STACK、VAR、CANDS の五つの各領域をそれぞれ割り当てる。また、これ以外にすべてのプロセスユニット間で共有する DNET、CR とごみ集めで用いられる参照テーブルを置く。

3 ごみ集め

動的記憶管理を行っているなどで、明示的に領域を解放できないような場合、使用されなくなったセルは未使用のまま放置されることになる。このように放置されているセル(ごみ)を再利用するために回収することをごみ集めと呼ぶ。このような処理を行うことにより、メモリといった限りある資源を再び有効に活用することが出来るようになる。

ごみ集めは OS や多くの言語処理系、例えば Lisp に代表される記号処理言語の処理系など非常に多くで用いられる重要な技術である。さらに、効率の良いシステムを実装するのはあまり容易では無いため、古くから研究されている分野でもある。しかし、並列ごみ集めなどマルチプロセッサを前提にしている研究は比較的最近になってからの研究で、現在もさまざまな研究が行われている。

ごみ集めのアルゴリズムとしては、大きく以下の2つに大別できる。

- リファレンスカウントによる方法
- トレースによる方法

4 並列ごみ集め

TRAM においてごみ集めの対象になるのは、CODE 領域に蓄えられるマッチングプログラムのみである。書換えにより変更されるのは部分的なマッチングプログラムであり、変更後のマッチングプログラムを空き領域に格納しそれを指すようにポインタが変更される。この操作によって変更前の部分的なマッチングプログラムはどこからも参照されないセルとなることが分かる。これがごみとなるのである。TRAM でのごみ集めの方法は、単純なコピー方式を用いていた。

Parallel TRAM でのごみ集めの基本的な方法は TRAM の場合と同様にコピー方式のごみ集めとなっている。これは、

- セル(マッチングプログラムの各要素)の長さが不定なため詰め替えの動作が必要
- ごみに比べ必要なセルの量が少ない

という大きな2つの理由からである。

また、Parallel TRAM の各処理ユニットの独立性に注目し、分散メモリ型マルチプロセッサで用いられているような分散型ごみ集めのうちコピー方式を基本としたごみ集めの採用を考えていく。Parallel TRAM の CODE 領域は処理ユニットごとに個別に持ち、各ユニットは独立して書換えを行っているという独立性によく合うものと考えられる。

しかし、Parallel TRAM では外部参照が起きる場合がある。これを解決するために外部参照のテーブルによる外部参照を実現する。

ただし、ごみ集めを行う際に参照テーブルを走査する場合など、collector に対する参照にロックを掛け排他制御を行う必要がある。しかし、ロックを掛けるという処理は一般に非常に重いものでありなるべく避けることが望ましい。そこでこの重い排他制御を減らすための工夫も行っている。

5 結論

本研究では、並列ごみ集めを Parallel TRAM に対し設計・実装を行った。また、これを6台の Ultra SPARC を持つ Ultra Enterprise 4000 system に実装し、性能の評価を行った。実装に用いた言語は C 言語である。得られた結果は以下のようになる。

- グローバルな同期を行わずにローカルなごみ集めを行う並列ごみ集めが出来るようになった。
- Parallel TRAM におけるごみ集めのオーバヘッドを減らすことが出来た。
- これにより最大約 20% の性能の改善がみられた。