

INCREMENTAL CYCLIC GARBAGE COLLECTION FOR MULTI-COMPUTERS

4V-11

前田宗則 小中裕喜 石川裕 友清孝志 堀敦史
新情報処理開発機構つくば研究センター

1 はじめに

トレーシングガーベッジコレクション (tracing-GC) は、システム中のオブジェクトのアクセス可能性をルートからの到達可能性によって判断する。全ての到達可能なオブジェクトは印付けされ、その処理の終了後に印付けされなかったオブジェクトをゴミとして回収する。こういったゴミ回収の枠組みを分散メモリを持つ並列マシン（以下、単に分散並列マシンと表記する）に適用するとき、印付けのためのメッセージがプロセッサ間でやりとりされることになる。通常ローカルメモリに比較して分散メモリのアクセスはオーバーヘッドが大きいので、分散並列マシンに対する GC アルゴリズムは、できるだけ印付けメッセージを減らすことが求められる。

本稿で提案される GC アルゴリズムは、ゴミと疑わしきオブジェクトから到達可能性閉包の部分集合を求め、その集合外のオブジェクトからその集合への参照が存在しないことを同定することにより、サイクリックなりモート参照を含むゴミオブジェクトを回収する。アクセス不能なオブジェクトを辿る点で、アクセス可能なオブジェクトを辿る従来の tracing-GC とは本質的に異なる。

2 関連研究

これまでに提案された分散並列マシン用の GC アルゴリズムのなかで、POOL-T GC 方式 [1] は、リモートオブジェクトの印付けを印付けメッセージによって行うグローバル GC (GGC) を提案している。本アルゴリズムの特徴は、各プロセッサごとにそのメモリ上の印付けを担当するガーベッジコレクションプロセス (GP) の導入と単純な GGC 終了判定プロトコルを提案した点にある。SGP 方式 [4] は、リモート参照を含む任意のオブジェクト構造を GC 対象とすることを諦めることで、印付けメッセージをやりとりしない。また、各プロセッサが他のプロセッサと独立にローカルなゴミを回収するローカルガーベッジコレクション (LGC) を提案している。鎌田の方式 [5] は、POOL-T GC よりもメッセージ数を大幅に減少させた GGC に加えて、さらにコスト高な GGC の頻度を下げるために LGC を併用している。

3 “赤と黒” ゴミ回収アルゴリズム

我々が提案する“赤と黒”ゴミ回収方式 (RBGC) は、分散プロセッサ毎に GP を持ち、LGC と GGC を併用してゴミ回収を実施する。

- LGC 処理において、ローカルに到達可能でなく、かつリモート参照が存在しないオブジェクトを回収する。これは各 GP が独立に実施する。
- GGC 処理において、サイクリックなりモート参照を持つゴミオブジェクトを回収する。これは関係する GP が同期して実施する。

RBGC の GGC アルゴリズムは、以下のような処理手順を踏む。i) ローカルに到達可能でなくリモートに参照されているオブジェクトの少なくとも1つを GGC の開始点として選択する。ii) これ (ら) から到達可能なオブジェクトを全て赤に塗りつぶすことにより、赤オブジェクトのみからなる集合 R を求める。iii) 集合 R に含まれるオブジェクトのうち集合 R 以外のオブジェクトから参照されているもの、およびそれらのオブジェクトから到達可能な R 中のオブジェクトを除く。これは赤を脱色する操作によってなされる。ここで得られた新しい集合を G とする。iv) 集合 G のオブジェクトをゴミとして回収する。図1は、印付け範囲の観点から従来の tracing-GC と RBGC を比較した模式図である。

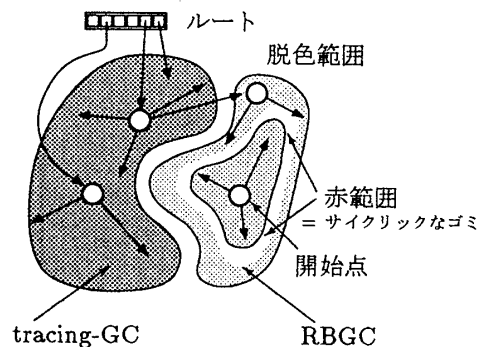


図1: 印付け範囲で比較した2つのアルゴリズム

処理 iii で脱色の開始対象となるオブジェクトは、mutator からの送信メッセージが未到着であるものと集合 R 外から参照されているものとなる。ある赤オブジェクトが集合 R の外から参照されているならば次の命題が成り立つ。

1. 任意の分散メモリ上において、赤でないオブジェクトから赤オブジェクトへのローカル参照が存在する。

2. ある赤オブジェクトへのリモート参照を保持する参照元オブジェクトのうち赤でないものがある。

命題1は、赤オブジェクトを含む分散メモリ上でローカルルートからのLGCを行なうことで、また、命題2は、処理iiの赤伝播によって各赤オブジェクト毎に参照元赤オブジェクトからのリモート参照数を数え挙げることでそれぞれ検証できる。

GGCによるゴミ回収をインクリメンタルに行なうためには、LGC実行後に非赤オブジェクトから赤オブジェクトに対するローカル参照が新たに生成された場合、および赤伝播でトラバースしたリモート参照がコピーされた結果、数え挙げられた参照元赤オブジェクトからのリモート参照数が実際の参照数と一致しなくなる場合の処理を新たに必要とする。どちらの場合においてもR外から参照されている赤オブジェクトが必ず脱色されることを保証しなければならない。これは、ゴミでないオブジェクトが誤って回収されないようにするために重要である。

4 RBGCの実現方式

RBGCは、SGP方式と循環参照カウント方式[3]を実現の基礎とする。各分散メモリ上には、入力されたリモートポインタを記録する外部参照表(ERT)と輸出されたローカルポインタを記録するオブジェクトアドレス表(ODT)が存在している。LGCを行なうときには、ルートとODTからトレースを行なえばよい。また、ODTとERTの各エントリは、オブジェクトの参照数を保持しており、ポインタの輸出に当たっては、重み付き参照カウント方式[2]と類似の手続きを踏む。

GGCの処理iiにおける赤伝播は、開始点としてODTから指されるオブジェクトを選び、それ(ら)のオブジェクトの保持する赤ビットをONにする。赤オブジェクトからローカルに到達可能なオブジェクトは全て赤ビットが設定される。リモート参照先に対しては、対応するERTエントリに記録されている参照重みを通知しリモートオブジェクトを赤にするためのメッセージを参照先プロセッサのGPに送信する。受信したGPは、オブジェクトを赤にするとともにそのオブジェクトから到達可能なオブジェクトに赤を伝播する。通知された参照重みは、処理iiiにおいて命題2をチェックするために使われる。赤伝播の終了判定は、POOL-T GCの終了判定と同様なプロトコルで行なえる。

処理iiiの脱色伝播は、赤オブジェクトのビットをOFFにするとともに、リモート参照先に対しては、そのオブジェクトを脱色するためのメッセージを相手先GPに送信する。受信したGPは、脱色操作を到達可能なオブジェクトに適用する。処理iiiの終了条件は、赤オブジェクトの存在する全てのプロセッサが少なくとも1度LGCを実行したことと、脱色要求メッセージがすべて処

理されたことが保証されることである。インクリメンタルなゴミ回収において、脱色もれが生じないように赤オブジェクトの到達可能性のスナップショットを取る。これは、LGC実行後から処理iiiの終了時まで赤オブジェクトに対してのみライトバリア操作[6](赤ライトバリア操作と呼ばれる)を行なうことで実現できる。

5 おわりに

RBGCのGGCは、全オブジェクト空間ではなく、開始点オブジェクトから到達可能な部分空間のみをゴミ回収の対象とする。処理がより局所的に行なわれるという性質は、分散システムにとって望ましいものである。また、インクリメンタルなゴミ回収が行なえるので実時間を意識したユーザプログラムにも適している。

効率に関しては、処理iにおいて適切なオブジェクト、すなわち、サイクリックなゴミオブジェクトが選択され、ゴミオブジェクト集合Gに含まれるリモート参照の数がアクセス可能なリモート参照の数よりも少ないならば、tracing-GGCより低オーバーヘッドで集合R中のオブジェクトを回収できる。逆に、赤に塗ったオブジェクトが全て脱色されるような場合においては非常に効率が悪化する。それゆえ、開始点となるサイクリックなゴミオブジェクトをいかに高い頻度で見つけられるか、無駄な赤伝播をいかに防ぐかが実行効率を大きく左右する。無駄な赤伝播を防ぐためには、ローカルルートから到達可能なオブジェクトを早期に適用対象外とすることである。LGCとGGCは独立に行なえるが、これらの協調処理が実施された場合にはローカルには無駄な伝播を防ぐことができる。

効果的な赤伝播(脱色)戦略と効率的な赤ライトバリア操作は稿を改めて議論することにする。

参考文献

- [1] Lex Augusteijn. Garbage collection in a distributed environment. In *Lecture Notes in Computer Science*, Vol. 259, pp. 75-93. Springer-Verlag, 1987.
- [2] D. I. Bevan. Distributed garbage collection using reference counting. In *Lecture Notes in Computer Science*, Vol. 259, pp. 176-187. Springer-Verlag, 1987.
- [3] Richard E. Jones and Rafael D. Lins. Cyclic weighted reference counting without delay. In *Lecture Notes in Computer Science*, Vol. 694, pp. 712-715. Springer-Verlag, June 1993. (5th International PARLE Conference Proceedings).
- [4] Marc Shapiro, David Plainfossé, and Olivier Gruber. A garbage detection protocol for a realistic distributed object-support system. Rapport de Recherche 1320, INRIA, November 1990.
- [5] 鎌田十三郎, 松岡聡, 米澤明憲. 超並列計算機上の高効率な大域的ガベージコレクション. 並列処理シンポジウム JSP'94 論文集, pp. 33-40. 情報処理学会, May 1994.
- [6] Taiichi Yuasa. Real-time garbage collection on general-purpose machines. *Journal of Systems and Software*, No. 11, pp. 181-198, 1990.