

# Partial Marking GC

田中 良夫

松井 祥悟

4V-9

慶応義塾大学理工学研究科

神奈川大学理学部

## 1. はじめに

現在並列 GC で用いられているアルゴリズムのほとんどは Snapshot-at-Beginning 型アルゴリズム (SB 型)<sup>[1]</sup>である。しかし、SB 型にはゴミセルの回収効率が停止型に比べて約 1/2 になってしまうという欠点がある。そのため GC を並列に行なっても、アプリケーションの実行時間や実時間処理の能力に関してあまり良い結果が得られない。本研究の目的は並列 GC の実用化であり、本論文では SB 型の効率改善の方法として Partial Marking GC (PMGC) を提案し<sup>[2, 3]</sup>、その効果および効率に関する解析について報告する。PMGC は、SB 型に世代別 GC の考え方を導入した GC である。SB 型の通常の GC サイクルの間に、一部のセルに対してだけ印づけを行う partial marking を含む GC サイクルを挿入する。この partial marking では直前の GC サイクルで生き残ったセルはその回の GC サイクルでも生きていますとみなし、印づけを行わない。PMGC は、印づけにかかる時間を短縮することにより平均的な GC 時間を短縮し、並列 GC の回収効率を上げる効果的な手法である。

## 2. SB 型アルゴリズム

SB 型は「ルート挿入が行なわれた時点で生きていたセルおよび、印づけフェーズの間に生成されたセルは直後の回収フェーズでは回収しない」という考えに基づいて GC の処理の正当性を保証している。SB 型は簡潔であり、実装も容易であるが、逆に考えれば「ルート挿入が行なわれた時点で生きていたセルと、印づけフェーズの間に生成されたセルは、すぐ

にゴミになったとしても、直後の回収フェーズでは回収されず、もう一度印づけフェーズを経た、次の回収フェーズで回収される」ということになる。世代別 GC の基本的な概念である「ほとんどのセルは生成後間もなく死んでしまう」ということを考えれば、ゴミセルの多くがすぐに回収されないことが SB 型の効率低下の原因となることが予想できる。

## 3. Partial Marking GC

PMGC は、SB 型に世代別 GC の考え方を導入した GC である。PMGC は印づけにかかる時間を短縮することにより生成後すぐに死んでしまったセルをより早く回収し、並列 GC の回収効率の上昇をはかる。PMGC では、full cycle と partial cycle の二種類の GC サイクルを用いる。full cycle は SB 型の GC サイクルと同じであるが、partial cycle では、前回の GC サイクルで生きていたセルは生き続けているとみなし、それらのセルには印づけを行なわない。その結果、そのサイクルでの印づけは直前の GC サイクルとの差分部分 (前回の印づけフェーズと回収フェーズの間に生成され、ルート挿入の時点で生きていたセル) だけを印づけする部分的な印づけ (partial marking) となる。この部分的な印づけでは印をつけるセルの数が極めて少なく、この印づけは短時間で終了することになり、前回の回収フェーズで回収できなかったゴミセルをすぐに回収することができる。印づけにかかる時間が短縮され、並列 GC の回収効率も改善される。

## 4. 効率の解析

Hickey らの解析に基づいて、SB 型および PMGC の効率の解析を行なう。ただし、セル空間の大きさを  $N$  とし、生きていたセルの数を  $L$  とする。フリーリストが空でないとき、リスト処理プロセス (LP) は毎  $r$  単位時間に 1 個のセルを生成し、同じ割合でゴ

Partial Marking GC

Yoshio TANAKA  
Shogo MATSUIGraduate School of Science and Technology, Keio University  
Faculty of Science, Kanagawa University

ミを出すものとする。従って生きているセルの総数  $L$  は常に一定となる。GC については毎  $m$  単位時間に 1 個セルに印をつけ、毎  $s$  単位時間に 1 個セルを走査するものとする。また、「1 回の GC サイクルの間に LP が出すゴミは black か off-white のどちらかである。そのうち、black のゴミが占める割合を  $b$  で表す」と仮定する。

SB 型では新たに生成されたセルのタグは off-white のままであるので、それらのセルに印をつける必要はない。LP が一度も待たされることがなければ、1 回の GC サイクルの間に作られるセルの総数(このセルの個数を  $G^{crit}$  とする)は一定の値となる。 $G^{crit}$  に関して SB 型で LP が待たされずにすむための条件を求めると、次のようになる。

$$G^{crit} \leq \frac{N-L}{2} \quad (1)$$

また、PMGC において LP が待たされずにすむための条件は

$$G^{crit} \leq \frac{\frac{r-s}{r}N-L}{1+\frac{m+r}{r}b} \quad (2)$$

となる。求められた条件式に Hickey らの解析で用いられているデータ ( $N = 500k, L = 300k, s = 1\mu, m = 5\mu, r = 20\mu$ ) をあてはめてみると、SB 型の場合には  $G^{crit}$  が 100k を越えるまで stable であることになる。PMGC では  $b$  の値によって stable であるための条件が変わってしまう。 $b$  の値は実行するアプリケーションに依存するパラメタであるが、「通常、新たに生成されたオブジェクトの大部分 (80 から 98%) は数 10 キロバイトのアロケーションが行われる間に死んでしまう」<sup>[1]</sup> という報告もなされており、 $b$  の値はおよそ 1 割から 2 割程度であることが予測できる。 $b$  の値を 2 割と仮定すれば PMGC での stable の条件は  $G^{crit}$  が 140k を越えるまでであり、 $b$  の値を 1 割と仮定すれば  $G^{crit}$  が 156k を越えるまで stable であることになる。

## 5. 実験

PMGC は OMRON LUNA-88K ワークステーション上で実装された。並列 GC の効率を評価するために、GC 率  $G$  を「停止型 GC で実行した場合に、全体

の何割が GC に費やされているか」、改善率  $I$  を「並列 GC によってどの程度実行時間が短縮されるか」を表す値とする。 $G$  および  $I$  は実験より求めることができる。 $G$  と  $I$  のグラフを図 1 に示す。グラフの単調増加部分が実時間処理可能な  $G$  の範囲を示す。点線が理想的な直線である。SB 型に比べて PMGC の効率が大きく改善されていることが分かる。

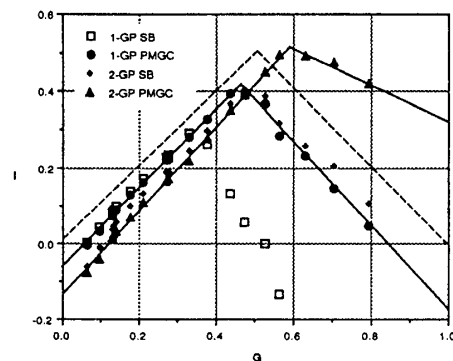


図 1: GC 率と改善率

Fig.1: GC ratio vs. Improvement Ratio

## 6. 結論

本論文では SB 型アルゴリズム改善の方法として PMGC を提案し、その効率を解析した。実験結果は解析結果が正しいことを示した。PMGC は並列 GC の実用化に向けての有効な手段となる。

## 参考文献

- [1] Wilson, Paul R.: "Uniprocessor Garbage Collection Techniques", Lecture Notes in Computer Science, 637, Springer-Verlag (1992).
- [2] Tanaka, Y. et al.: "Parallel Garbage Collection by Partial Marking and Conditionally Invoked GC", Proc. of the International Conference on PaCT-93, 2, (Sep. 1993).
- [3] Tanaka, Y. et al.: "Partial Marking GC", Proc. of the International Conference on CONPAR 94 - VAPP VI, (Sep. 1994) (to be appeared).