
発表概要

Java 環境における粗粒度インクリメンタル GC の設計

矢田 智紀[†] 前田 敦司[†] 山口 喜教[†]

マークスイープアルゴリズムに基づくインクリメンタルガベージコレクタは、停止時間をぐくわずかに抑えることが可能であり、リアルタイム応用に適しているが、CPU 時間が増加するという欠点がある。この性能低下の原因は、一括型 GC と比較してマークする時期が早いことに起因する mark/cons 比の悪化、GC ルーチン呼び出す回数の増加、ライトバリアのオーバーヘッドなどによる。これに対し、世代別ガベージコレクタは高い CPU 効率が得られ、また平均の停止時間は比較的短いために良いレスポンスが得られるが、旧世代領域の GC (メジャーコレクション) の際には長い時間計算処理が停止してしまい、リアルタイム応用には適さないという欠点がある。これら両手法の特徴をふまえ、本発表では、世代別ガベージコレクタをベースとした粗粒度インクリメンタル GC を提案する。これは、新世代領域 (マイナーコレクション) のたびに、インクリメンタルに旧世代領域のガベージコレクションを行うことで、世代別ガベージコレクタの高い CPU 効率を保ちつつ、リアルタイム性を確保する手法である。今回は、同アルゴリズムを IBM ワトソン研究所の JikesRVM 上に実装し、評価を行ったうえで、その有効性を示す。

Design of Coarse-grained Garbage Collection on Java Environment

TOMONORI YATA,[†] ATSUSHI MAEDA[†] and YOSHINORI YAMAGUCHI[†]

Incremental garbage collectors based on mark-sweep algorithms can minimize the pause time caused by garbage collection at the expense of increased CPU time. The main reasons for this performance degradation include: mark/cons ratio gets worse in incremental collectors because objects are marked earlier than in non-incremental counterparts, increase in number of calls to collectors routine, and write barrier overhead. On the other hand, generational collectors can achieve high efficiency and relatively short pause time in average. But it is not fully suitable for real-time application, since they occurs some long pauses by garbage collection for old generation space (major collection). In this presentation we propose a new GC algorithm which is called Coarse-Grained Incremental GC based on generational garbage collector. This algorithm incrementally reclaims old generation space every time a minor collection is performed, and it can not only achieve high efficiency but also ensure real time response. We demonstrate effectiveness of this algorithm by implementing and evaluating on JikesRVM which is Java VM for research developed by IBM T.J. Watson Research Center.

(平成 17 年 8 月 4 日発表)

[†] 筑波大学システム情報工学研究科
System and Information Engineering, University of
Tsukuba