
発表概要

大容量キャッシュに向く圧縮型ガーベッジコレクションについて

藤川 国雄[†] 新田 寛[†] 寺島 元章[†]

比較的大きな容量の二次キャッシュを搭載した計算機上の Lisp 処理系について、その圧縮型のガーベッジコレクション (GC) のプログラム実行時間に対する効果を複写型 GC と対比して述べる。両者は停止回収型に属し、二世代のデータオブジェクトを効果的に処理する世代別 GC (二世代 GC) である。この圧縮型 GC はポインタ補正などに高速化のための改良や、ヒープの消費域を局所化する対策が行われている。圧縮型は複写型よりも再配置 (移動) オブジェクト量に対する処理時間がかなり長い、このことがプログラムの実行に要する総処理時間の長短に必ずしも直結しないことを示す。総処理時間に対する GC 時間の比率、GC が行うデータオブジェクトの再配置やヒープの使用形態の違いなどが影響するからである。比較実験では、二次キャッシュ容量がそれぞれ、512 KB, 4 MB, 8 MB の同系のアーキテクチャである計算機上にそれぞれの GC を持つ Lisp 処理系を実装し、規模の異なるプログラムを実行した。複写型 GC は総処理時間にほとんど差異がないが、圧縮型 GC は比較的大きなキャッシュ容量で総処理時間の短縮が図られ、時間的に優れたプログラムが多いという結果が得られた。これは圧縮型 GC の利点である位置に関する局所性の保存と固定容量記憶に対する効率性の良さに起因するものと考えられるが、こうした解析結果についても述べる。

Mark-and-compact Garbage Collection for Large-size Cache Memory

KUNIO FUJIKAWA,[†] HIROSHI NITTA[†] and MOTOAKI TERASHIMA[†]

In this presentation, we describe a good effect on program execution time that is produced by the garbage collection (GC) of a mark-and-compact type being embedded in our Lisp processing system running on machines with relatively large-size cache memory loaded, and also describe an effect by the GC of a copying collection type. Both of the GC belong to the great group called stop-and-collect GC and the group of generational GC that processes two generation of data objects effectively. The GC of a mark-and-compact type, or mark-and-compact GC, proposed here has refinement that improves in time of its process such as pointer adjustments and a mechanism that localizes the consumption of a heap. The mark-and-compact GC takes far longer time than the GC of a copying collection type, or copying collection GC, to process the unit of data objects being relocated (moved), but this fact does not necessarily apply to the total time of program execution. There are many factors that affect the total time such as the ratio of GC time to the total time, relocation of data objects and heap utilization in which both GC differ much. We test performance of Lisp programs running on 512 KB, 4 MB and 8 MB cache machines. The copying collection GC has few effect on the total time, but the mark-and-compact GC shortens the total time in case of relatively large-size cache and makes the total time of many programs shorter than the copying collection GC. This fact may result from the merits of the mark-and-compact GC: locality of data objects in the heap and efficiency of the heap utilization.

(平成 15 年 3 月 18 日発表)

[†] 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of
Electro-Communications