

発表概要

スタックベース GC の提案と スクリプト言語 Lua における評価

小室 直^{†1} 阿部 公輝^{†1}

現在多くのスクリプト言語処理系に Garbage Collection (以降 GC) が組み込まれている。GC による自動メモリ管理はスクリプト言語に大幅な自由を与えるが、処理時間とメモリ使用量が増加し、リソースが限られた環境において動作するビデオゲームのようなアプリケーションでは問題となる。本論文では動的なリージョン操作によりオブジェクトを管理するスタックベース GC を提案し、スクリプト言語処理系に組み込む。リージョンベースメモリ管理とはオブジェクトの確保と解放をリージョンという単位で区切って管理する手法である。オブジェクト領域は GC 管理領域に格納されるため、オブジェクトスタックから GC 管理領域への移動のコストが小さい。またループごとにリージョンを解放するようにするため、解放されたオブジェクト領域がうまく再利用され、キャッシュによる高速化が見込める。評価実験により GC が必要なプログラムに対してスタックベース GC は GC 処理時間が大きく減少させることが確認できた。またリソースの少ない Nintendo-DS における評価実験では、従来の GC ではリソース不足により正常動作しないプログラムが動作することを確認した。

A Proposal of Stack-based Garbage Collection and Its Evaluation in Scripting Language Lua

SUNAO KOMURO^{†1} and KOKI ABE^{†1}

We propose a new scheme of garbage collection (GC) and present its evaluation results through the implementation of the scripting language Lua. Conventional GC schemes embedded in scripting language processors are often delayed by long pause times, and programs requiring real-time responses need to be carefully coded. There are currently several types of incremental GCs that solve this problem, but in doing so they sacrifice the GC's throughput. In this paper, we propose a stack-based GC scheme based on Corry's "Optimistic stack allocation for java-like languages." This scheme prepares a stack that stores objects. The stack frame is segmented per loop and deallocated with

fine granularity. The pattern of memory allocations and deallocations has a LIFO structure in which objects that are spatially close and placed temporally close as well, resulting in enhanced efficiency of cache usability caused by the regular pattern of memory accesses. Because memory is deallocated at each end of the loop, the GC cost is easily predicted. A periodically invocated finalizer leads to the fine granular deallocation of resources such as file descriptors. We implemented the proposed GC in a Lua processor and observed that the speed is improved up to 1.67 times more than the conventional GC and that the memory usage is reduced by up to 83%. We ported Lua to a Nintendo-DS and observed that programs unable to run with conventional GC can be executed with the stack-based GC.

(平成 22 年 6 月 14 日発表)

^{†1} 電気通信大学情報工学専攻

Department of Computer Science, The University of Electro-Communications