
発表概要

Ruby のメモリ管理の改善

笹 田 耕 ^{†1}

プログラミング言語 Ruby の処理系は、保守的マークアンドスイープガベージコレクション (GC) による自動メモリ管理機構を有する。本稿では、より高性能、高効率なメモリ管理を実現する手法について検討する。まず、ヒープサイズをこれまでよりも大きくとることで GC オーバヘッドを削減する方針をとる。ヒープ拡大により、オブジェクトフラグメンテーションが発生するが、これを防ぐためにいくつかの手法について検討する。また、マーク時にマークカウンタを用いることで、スイープ時にブロック単位で領域を開放することを可能にし、スイープ時間の削減を目指す。また、他プロセスと協調したヒープサイズの調整を行うため、ページアウトを監視する手法についても述べる。また、これらを実現するために `mmap()` システムコールなど、OS 依存の機構を用いてメモリ管理機構を一新する。本発表では、これらの手法の検討結果および実装方法について述べ、予備評価の結果を示す。

Improving Memory Management for Ruby

KOICHI SASADA^{†1}

The interpreter of programming language Ruby has a conservative mark and sweep garbage collection (GC) mechanism. In this paper, we consider about performance and efficiency improvement about memory management on the Ruby interpreter. At first, we take a course to expand heap area to improve memory management. Solutions against object fragmentation problems due to heap expansion are considered. We also introduce mark counter technique to reduce a sweep overhead. To make cooperative heap-size adjustment with other processes in same system, page out information are used. We implement memory management system with mechanisms depend on OS such as `mmap()` system call. In this presentation, we show our proposals, implementations, and result of preliminary evaluation.

^{†1} 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo