

発表概要

効率的なビットマップマーキングを用いた Ruby 用ごみ集め実装

中村 成洋^{†1} 松本 行弘^{†1}

Apache HTTP サーバのもとで提供される Web サービスのような、ひんばんにサブプロセスが生成される環境下では、現在 Ruby が採用しているマークビットをオブジェクトヘッダ内を持つマークスイープ方式のごみ集め実装は、copy-on-write によるメモリページ共有を疎外し、必要以上にメモリを消費する。本研究では、マークをオブジェクトヘッダ内から独立したビットマップに移すことによるメモリ消費量実行性能の変化を示す。効率的なビットマップマーキングのためにはオブジェクトポインタからビットマップの特定の位置の計算が必要である。一般的にはビットマップ位置をたとえば 1M バイト単位でアラインしておきアドレスをマスクすることでビットマップ位置を求めることが行われている。しかし、Ruby のように種々のプラットフォームで動作する言語処理系では、利用できるメモリ割当て API は malloc() しかないため、効率良くアラインされたメモリを確保する方法は知られていない。本研究では Ruby のオブジェクト配置方式を利用し、移植性を維持したまま、オブジェクトポインタから定数時間でビットマップ位置を計算する手法を提案する。また、二分検索を用いたビットマップマーキングと比較したマーキング性能の改善も示す。

do not have portable memory allocation API to obtain aligned memory region without wasting region. In this paper, we propose portable scheme to map from object pointers to corresponding bitmap table in constant time. We also show how much proposed bitmap marking improves performance, comparing bitmap marking method using binary search to obtain bitmap position from object pointers.

(平成 20 年 10 月 28 日発表)

Ruby Garbage Collection Using Efficient Bitmap Marking

NARIHIRO NAKAMURA^{†1} and YUKIHIRO MATSUMOTO^{†1}

Since mark-and-sweep garbage collection scheme, which Ruby interpreter uses modifies every living object, it suffers performance problems due not to utilize copy-on-write memory page sharing among processes, under the circumstances like web-services running under Apache HTTP servers. In this paper, we propose adding bitmap marking for Ruby's garbage collection. We show much memory usage and performance change by bitmap marking. For efficient bitmap marking, it is needed to calculate bitmap position from an object pointer. Prior art uses aligns heap memories and pointer masking to retrieve bitmap position from object pointer. But Ruby interpreter runs on various platforms, and we

^{†1} 株式会社ネットワーク応用通信研究所
Network Applied Communication Laboratory Inc.