

要 旨

サーバサイド向け JavaScript 処理系の ためのマークスイープごみ集めの 設計と実装

高松 裕樹

Web ブラウザの高機能化に伴って、Web ブラウザ上で動作する Web アプリケーションが普及している。Web アプリケーションは、Web ブラウザ上で JavaScript のプログラムとして動作し、Web サーバ上のプログラムと通信しながら高度なサービスを提供する形態をとる。近年、Web アプリケーションのサーバ側のプログラムの開発言語として JavaScript が普及してきている。そのため、サーバ上で動作する JavaScript (サーバサイド JavaScript) の処理系の高速化の研究が盛んに行われている。サーバ上では大規模なプログラムを動かすことが多いと予想される。その際にプログラムの処理時間に大きく影響を与えるのが、ガベージコレクション (以下、GC) である。GC とは、メモリ領域上のプログラムが利用しなくなった領域を回収して、再度利用可能にする処理である。GC 実行中はプログラムの処理が停止しているため、GC の時間はそのままプログラムの実行時間に影響を与える。本研究では、サーバサイド向け JavaScript 仮想機械である SSJSVM の GC について、他の処理系に比べて高速かどうか調べた。その結果、既存の SSJSVM の GC は他の処理系の GC に比べて遅いことがわかった。そこで、GC の性能改善を狙って SSJSVM にマークスイープ GC を実装した。本研究で実装した GC の性能を調べるために、既存の SSJSVM と実行時間を比較した。その結果、本研究の GC は SSJSVM の性能を改善しないことが分かった。さらに調査した結果、本研究の GC はメモリ割り当てが遅いことがわかった。

キーワード JavaScript, ガベージコレクション, マークスイープ, BoehmGC

Abstract

Design and Implementation of a Mark Sweep Garbage Collection on a Server Side JavaScript Virtual Machine

Yuki TAKAMATSU

With the high functionality of Web browsers , Web applications widely spread . Web applications are written in JavaScript and work on Web browsers . A Web application communicates with a program running on the server to provide rich services to the users . Recently , JavaScript spread as a language for server side programs of Web applications . Therefore , studies of speeding up JavaScript processing on servers are conducted flourishingly . On servers , it is expected that we often use large-scale programs . In this situation , it is garbage collection (GC) that greatly affect the execution times of programs . GC is a routine that reclaims memory areas that are no longer used so that the user program can reuse the areas . Because the execution of the program stops during GC , GC affects the execution time of the program directly . In this study , we investigated whether GC of a server side JavaScript virtual machine , SSJSVM , is efficient compared to other JavaScript virtual machines . As a result , we found that GC of SSJSVM was slower . Then , we implemented a mark sweep GC in SSJSVM aiming the performance improvement . To investigate performance of the GC that we implemented in this study , we compare the execution time of our SSJSVM with that of the existing SSJSVM . As a result , we found that our GC did not improve performance . Further investigation showed that memory allocation of our GC was much slower than that of the existing SSJSVM .

key words JavaScript , Garbage Collection , Mark Sweep , BoehmGC