

論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Techniques for Mixed-mode Virtual Machines to Improve Overall System Performance
著者(和文)	緒方一則
Author(English)	Kazunori Ogata
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:乙第4128号, 授与年月日:2016年4月30日, 学位の種別:論文博士, 審査員:増原 英彦,南出 靖彦,渡邊 治,首藤 一幸,脇田 建
Citation(English)	Degree:., Conferring organization: Tokyo Institute of Technology, Report number:乙第4128号, Conferred date:2016/4/30, Degree Type:Thesis doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

(論文博士)

論 文 要 旨 (和文2000字程度)

報告番号	乙 第 号	氏 名	緒方 一則
<p>(要 旨)</p> <p>プログラミング言語の実行時環境として、プログラムを仮想マシン (VM) のバイトコードに変換し、それをVM上で実行する手法が広く用いられている。この手法は、プログラム言語仕様と実行時環境の実装を分離することで、移植性を担保しつつ、マシン依存の最適化を適用可能であるなどの利点がある。</p> <p>VMを用いる実行時環境の中でも、動的コンパイラとインタプリタを組み合わせるmixed-mode VMは、定常状態の実行速度改善とプログラム起動時間の短縮を両立する手法として広く用いられている。例えば、OpenJDKやIBM J9 Java VMはmixed-mode VMである。</p> <p>本研究は、企業の業務システムで用いられる製品品質VMの性能改善手法を提案する。本番環境で求められる性能改善には、プログラム起動時間や負荷パターン変動への応答時間の短縮、複数プログラムの同時実行による総合性能の改善、問題発生時のプログラム停止時間の短縮など、定常状態の実行速度以外の性能指標も含まれる。動的コンパイラの改善による定常状態の実行速度改善は既に多くの研究が行われているが、本研究はそれらの性能指標を改善により、総合的なシステム性能改善を目的とする。</p> <p>プログラム起動時間や負荷パターン変動への応答時間の短縮のためには、インタプリタの実行速度改善が重要である。動的コンパイラは同一コードを複数回実行後にコンパイルするので、プログラム起動時や負荷パターン変動時などで新たに実行されるコードはインタプリタで実行される。そのため、インタプリタの実行速度改善により起動時間や応答時間を改善できる。</p> <p>本研究では、インタプリタがバイトコード命令列を読み込むためのメモリアクセスの冗長性に着目した。この冗長性は、バイトコード命令セットの特徴である可変長フォーマットと短い平均命令長のために、ワード境界に整列されていないメモリアクセスを同一メモリ・ワードに対して繰り返すことが原因である。</p> <p>この課題を解決するために、バイトコード命令の読み込みをワード境界に整列されたメモリアクセスで行い、読み込んだバイトコード列をレジスタにキャッシュした。キャッシュしたワード内における、実行中のバイトコード命令のバイト位置判定を避けるために、命令のバイト位置ごとにカスタマイズしたバイトコード・ハンドラを用意して、インタプリタを最大約30%高速化した。</p> <p>複数プログラムを用いた総合性能の改善のためには、1台のマシンでより多くのプログラムを実行してCPU使用率を常に高い状態に維持し、単位時間あたりの処理量を増やすことが重要である。そのためには、個々のプログラムのメモリ使用量を削減し、多数のプログラムを実行してもメモリ不足に陥らないようにする必要がある。</p> <p>本研究では、JavaVMのメモリ使用状況を詳細に解析し、メモリ使用量削減のためにはJavaヒープ以</p>			

外のメモリ領域も無視できないほど多いことを示した。さらに、JITコンパイラ作業領域のメモリ割当パターンがJava VM内の他のコンポーネントのパターンと大きく違うために、解放された作業領域の大半がシステム・ライブラリ内のフリーリストに滞留し、物理メモリを無駄に消費していることを発見した。この無駄なメモリ消費を削減する手法として、JITコンパイラ内部のメモリ管理コンポーネントに **madvise** システムコール呼び出しを追加することで、無駄な物理メモリ消費の約78%を削減した。

問題発生時のプログラム停止時間の短縮のためには、デバッグにかかる時間を短縮する必要がある。しかし、本番環境ではデバッガ内でプログラムを実行する手法は使えず、本番稼働中に記録した情報やダンプファイルのみでデバッグする必要があり、プログラム開発中のデバッグより困難である。また、本番稼働中に記録するデバッグ情報はできる限り少なくし、プログラム実行速度の低下を避ける必要がある。

本研究では、特にデバッグが困難なJITコンパイラの動作を本番環境以外で再現しデバッグを補助する機能を、少ないオーバーヘッドで実現する方法を提案する。JITコンパイラのデバッグが難しいのは、コード最適化に使用可能な実行時情報の内容が、JITコンパイラが起動されるタイミングによって変化するために、プログラムを再実行するだけでは問題を再現できないためである。問題を確実に再現するには、本番稼働中にJITコンパイラが使用した実行時情報を記録しておく必要があるが、実行時情報を全て記録すると実行速度が大きく低下する。

この課題を解決するために、JITコンパイラがコード最適化に利用する実行時情報を、プログラム実行中に値が変化する可能性のあるものと変化しないものに分類し、前者のみを実行中に記録し、後者はJava VMのプロセス・ダンプを用いて保存することで、プログラム実行中の記録オーバーヘッドを本番稼働中でも有効化できる程度まで削減した。

これらの技術により、製品品質VMにおける定常状態の実行速度以外の性能の改善することで、本番環境の総合的なシステム性能を改善した。

備考：論文要旨は、和文2000字と英文300語を1部ずつ提出するか、もしくは英文800語を1部提出してください。

Note：Thesis Summary should be submitted in either a copy of 2000 Japanese Characters and 300 Words (English) or 1 copy of 800 Words (English).

注意：論文要旨は、東工大リサーチリポジトリ (T2R2) にてインターネット公表されますので、公表可能な範囲の内容で作成してください。

Attention: Thesis Summary will be published on Tokyo Tech Research Repository Website (T2R2).

(論文博士)

論 文 要 旨 (英 文)

(300語程度)

(Summary)

報告番号	乙 第	号	氏 名	緒方 一則
------	-----	---	-----	-------

(要 旨)

This dissertation presents techniques to improve the performance of mixed-mode virtual machines (VMs), which use both an interpreter and a dynamic compiler. Such VMs are widely used for implementing runtime systems of programming languages and have good steady-state performance and short startup times.

Production systems can be characterized with various performance metrics other than the steady-state one, such as the startup time and response time to changes of program behavior, total performance of multiple programs in a system, and the time to debug a problem in a runtime system.

The first contribution of this dissertation is a technique to increase the execution speed of an interpreter. In a mixed-mode VM, the interpreter executes a piece of code until it is dynamically compiled, and most of the code in a newly started program and the code executed just after changes of program behavior are not compiled. We improved interpreter performance by up to 30% by eliminating redundant memory loads for fetching bytecode instructions.

The second contribution is a technique to reduce the inefficiency caused by a dynamic compiler's memory allocation pattern. Efficient memory usage helps to avoid thrashing and keeps the total performance of multiple programs good even under high loads. We performed in-depth analyses of Java memory usage and found a large amount of unused memory in the system library's free list. We reduced 78% of its physical memory consumption by explicitly releasing its page frames.

The third contribution is a technique to reduce the overhead of the trace-and-replay mechanism of a dynamic compiler. A dynamic compiler is very difficult to debug because simply rerunning the same program does not necessarily reproduce the problem. We devised a technique to implement a trace-and-replay functionality for the dynamic compiler by recording all inputs to the compiler with a small overhead. We utilized a process dump to reduce the amount of data recorded during the execution of a program.

備考：論文要旨は、和文2000字と英文300語を1部ずつ提出するか、もしくは英文800語を1部提出してください。

Note : Thesis Summary should be submitted in either a copy of 2000 Japanese Characters and 300 Words (English) or 1copy of 800 Words (English).

注意：論文要旨は、東工大リサーチリポジトリ(T2R2)にてインターネット公表されますので、公表可能な範囲の内容で作成してください。

Attention: Thesis Summary will be published on Tokyo Tech Research Repository Website (T2R2).