論文の内容の要旨

論文題目　Adaptive Optimization of Language-Specific Overhead for Java Virtual Machines
（仮想機械上での言語機能固有オーバーヘッドの適応的最適化に関する研究）

氏名　　小笠原　武史

本論文はプログラムの開発効率を上げるためのプログラミング言語の機能が実行時に引き起こす性能オーバヘッドをJava実行環境で削減する手法を提案する．堅牢性，移植性，マルチプロセッサ上でのスケーラビリティを高めるためにJavaは例外処理，浮動小数点演算の精度保証，マルチスレッドを提供している．Javaが様々な分野で主要なプログラム開発言語として利用されるにつれ，こうした機能が多くのプログラムで利用されるようになった．そのためJava実行環境がプログラムを最適化してこれら言語機能を利用する際の性能オーバヘッドを削減することはプログラムの性能向上の鍵となる．

最適化のために，言語機能を利用するコードの実行時特性に適応した手法を用いる．標準ベンチマークや実環境を反映した実験プログラムを用いて言語機能を実装するコードの挙動を分析し，それを基にコードの実行時パスに重点をおいてJava実行環境で最適化するアプローチをとる．例外処理コストに対するアプローチは，実行時ライブラリにより例外処理を頻発させるコールスタックを収集し，ジャストインタイム（JIT）コンパイラが収集されたコールスタックについて例外処理の複雑な処理を単純分岐に変換する．実際に処理される例外のみに適応した動的最適化を行い，例外処理を行わない実行パスには余分なペナルティがない．浮動小数点演算精度保証コストに対するアプローチは，JITコンパイラが１つの精度モードでより長く実行するコードを生成する．メソッドに対しメソッドの呼出サイトと同じ浮動小数点精度モードを持つコードを必要に応じて生成する．また実行されるコードのプロファイル情報に基づきモード切替が実行頻度の高

いコードで起きないようにする. これにより浮動小数点モードを持つプロセッサ上で単精度演算と倍精度演算が混在するプログラムを実行する際の精度保証コストを減らすことができる. スレッド同期コストに対するアプローチは, 共有オブジェクトのロックが同じスレッドから続きやすいという特性を利用してロックはその解放後も最後に取得したスレッドが仮所有する. スレッドがロックを仮所有する限り，ロック操作とそうでない操作の直列化によるコストを削減できる. 別スレッドがロックを取得すると所有権はそのスレッドに移る.

提案した最適化手法が言語機能を利用したJavaプログラムの性能をどの程度改善できるかについて評価結果を示す．例外処理と精度保証の最適化の評価において提案手法をIBM Java JITコンパイラに実装した．業界標準であるSPECjvm98ベンチマークの７プログラム中４つで，既存の代表的手法より高い性能を確認した. 特に例外処理最適化はIBM Java実行環境の製品に組み込まれ, Javaの多くのプログラム製品の性能向上に貢献している．例外処理最適化はSPECjvm98ベンチマークの２つのプログラム_228_jackと_213_javacの性能をそれぞれ18.3%と13.8%改善した. 精度保証最適化は, メソッドインライン展開最適化の複数の方針および異なるアーキテクチャの複数のプロセッサで，既存手法より高い性能を確認した．SPECjvm98ベンチマークの２つのプログラム_227_mtrtと_222_mpegaudioの性能をそれぞれ8.36%と21.2%改善した．ロック最適化は，共有オブジェクトを利用するJavaサーバアプリケーションの２つの実世界シナリオにおいて，既存手法より102%と80%の性能向上を確認した.