
発表概要

機械語の生成を核とした Java コンパイラシステム

古橋 亮 慈[†] 森下 美奈子[†] 箭内 京子[†]
五十嵐 圭[†] 金山 邦明[†]
鈴木 貢[†] 渡邊 坦[†]

現在の Java 環境では、Just-in-Time (JIT) コンパイラにより実行時に、もしくは Ahead-of-Time (AOT) コンパイラにより実行前に機械語を生成する手法が一般化してきている。そこで、AOT コンパイラの一形態として、機械語の生成を核としたコンパイラシステムを提案する。JIT コンパイラや AOT コンパイラにおいては、Java 仮想マシンが本来持つ機能をどこまでコンパイラ側で実現し置き換えるかにより、多くの状況が考えられる。本システムにおいては、オブジェクトのメモリ配置および遅延束縛によるメソッド呼び出しをコンパイラで実装しているため、生成したコードは Java 仮想マシンがなくても実行できる。また、配列オブジェクトのアクセス高速化など、オブジェクトを直接扱うような最適化にも対応することができた。Java 環境におけるコンパイラは多様であり、ソースコードを入力して Java 仮想マシンの最適化されたバイトコードを得るもの、バイトコードを入力して機械語を得るもの、そしてソースコードから機械語を得るものが考えられる。本システムでは、将来的にそれらを統合的に扱えるようなリターゲット型の間接語を設計している。現在実装中のアーキテクチャは MIPS および SPARC である。本発表では、コンパイラを実装する過程における問題点とその解決法、および最適化の手法を示す。

Java Compiler System Based on Machine Code Generation

RYOJI FURUHASHI,[†] MINAKO MORISHITA,[†] KYOKO YANAI,[†]
KEI IGARASHI,[†] KUNIAKI KANAYAMA,[†] MITSUGU SUZUKI[†]
and TAN WATANABE[†]

In most Java environments, machine codes are generated by a Just-in-Time (JIT) compiler on the fly or by an Ahead-of-Time (AOT) compiler before execution. We propose a compiler system based on machine code generation as a kind of AOT compiler. In a JIT compiler or an AOT compiler, there are many varieties on how many features are covered by the compiler. In our system, object memory allocation and late-bound method call are implemented in the compiler, so that generated codes can be executed without a Java virtual machine. It can support optimizations handling object data structures such as efficient array accesses. There will be many kinds of compilers such as generating optimized byte code from source code, generating either machine code from byte code or from source code, etc. We designed retargetable intermediate codes in our system to integrate these compilers in the future. Now MIPS and SPARC architectures are supported in the machine code generation. This presentation describes problems, solutions, and optimization techniques for the implementation of the compiler.

(平成 13 年 3 月 27 日発表)

[†] 電気通信大学情報工学科

Department of Computer Science, The University of
Electro-Communications