

主 論 文 要 旨

報 告 番 号	甲 ㉞ 第	号	氏 名	千 葉 雄 司
主論文題名				
C 言語への変換による Java アプリケーションの高速化に関する研究				
内容の要旨				
<p>オブジェクト指向プログラミング言語 Java は生産性の高さなど様々な利点を持ち、産業界に普及しつつある。一方で Java で開発したソフトウェアには実行速度が遅いという欠点がある。Java アプリケーションの実行速度が遅いのは、Java アプリケーションが中間コードで書いてあり、汎用プロセッサでは直接実行できないからである。中間コードを実行する方法の1つにインタプリタがあるが、インタプリタによる実行は遅い。</p> <p>Java アプリケーションの実行を高速化する手段にコンパイラを使う方法がある。この方法では、コンパイラを使って中間コードを機械コードに変換し、プロセッサで直接実行可能にすることにより高速化を図る。コンパイラを使う方法の難点はコンパイラの開発コストにある。開発コストの軽減は組込み機器のようにコストへの要求が厳しい分野では特に重要である。開発コストを軽減する方法の1つに、コンパイラを Java から C 言語へのトランスレータと既存の C コンパイラから構成し、既存の C コンパイラに古典的最適化や機械コード生成を任せる方法がある。</p> <p>本論文では Java から C 言語へのトランスレータを実装するにあたって問題となる点を指摘し、その解決策を提案する。Java から C 言語へのトランスレータにとって最も大きな問題点の1つは、Java から C 言語へのトランスレータを使って作成した機械コードが、実行時に利用できるとは限らないことである。なぜなら Java から C 言語へのトランスレータは中間コードで書いてある Java アプリケーションを、プログラムの実行開始以前にあらかじめ機械コードに変換しておくが、機械コードの生成後に中間コードを変更した場合、古い中間コードから生成した機械コードは無効になるからである。</p> <p>従来の Java から C 言語へのトランスレータはこの問題を無視し、コンパイル後に中間コードを更新した場合でも無効になった機械コードを使う設計になっていたため、一部の Java アプリケーションを正常に実行できなかった。本論文ではこの問題を克服する方法として、実行時に機械コードを使う前に、その有効性を確認する手法を提案する。確認をとるまでの間は、インタプリタで中間コードを解釈実行するが、そのオーバーヘッドは必ずしも小さくない。そこで本論文では更に、有効性の確認にともなうオーバーヘッドを軽減する最適化技法として、クラス初期化検査の除去と、仮想メソッド呼出しの高速化を提案する。</p> <p>実用的な Java アプリケーションを集めたベンチマーク SPECjvm98 を用いて評価した結果、クラス初期化検査の除去により、平均 45%実行を高速化できることが判った。また、仮想メソッド呼出しの高速化により、最大 9.8%実行速度を改善できることが判った。</p>				