発表概要

頻出メソッド管理テーブルを用いたinvokeinterface命令の実行高速化手法

藤本勝平[†] 古関聰[††]
小松秀昭[††] 深澤良彰[†]

オブジェクト指向言語において，仮想的なメソッドのインデックスをつくり，このインデックスでメソッドの実体のテーブルを引くことで，仮想呼び出しの高速化を図る手法がある．この手法では，単一継承されたメソッドのインデックスはコンパクトに実装できるが，多重継承されたメソッドのインデックスは大きなメモリスペースを必要とするという問題があった．コンポーネント指向プログラムなどにおいては invokeinterface 命令が頻繁に呼ばれるため，この問題が多大な影響を及ぼす．そこで本手法では，Selected Selector Index Table（S2IT）という，サイズを制限したテーブルを用いる手法を提案する．この手法では，プロファイリングを行うことで頻繁に実行されるメソッドを調べ，それらのメソッドのみにインデックスを割り当てて，実用的なサイズのテーブルを構成する．一方，インデックスが割り当てられなかったメソッドの呼び出しは，通常の手法を通して行われる．これにより，大規模なアプリケーションにおいても，多重継承されたメソッドを効率良く呼び出すことができる．

Accelerating Invokeinterface Using Tables of Methods Which Call Frequently

Shohei Fujimoto,[†] Akira Koseki,[††] Hideaki Komatsu[††]
and Yoshiaki Fukazawa[†]

In an object-oriented language, there is an acceleration technique of virtual call to search virtual method table using the index created by virtual method. Single inheritance enables simple and efficient method dispatch. However, method dispatch tables of multiple inheritance and interfaces are considered too space-intensive to be practical. In the component-based programming, this problem introduces tremendous overhead because the interface method dispatch bytecode, invokeinterface, is called frequently. We propose a new optimization technique called “Selected Selector Index Table” (S2IT), which uses a small number of dispatch tables. At first, we find the methods frequently performed by profiling. Next, signatures of those methods are assigned unique small integer identifiers called selectors. Each class that implements interfaces maintains a practical number of dispatch table indexed by selector. If the method that isn’t indexed is called, the method dispatch is performed through the usual technique. Therefore, multiple inheritance and interfaces enable efficient method dispatch in large-scale application.

（平成 18 年 1 月 17 日発表）

† 早稲田大学大学院理工学研究科
Graduate School of Science Engineering, Waseda University
†† 日本 IBM 株式会社東京基礎研究所
Tokyo Research Laboratory, IBM Japan Ltd.