

---

**発表概要**

---

## バイトコードパターンマッチによる Java インタプリタの高速化手法

安江俊明<sup>†</sup> 仲 顕 照<sup>††</sup>  
小松秀昭<sup>†</sup> 深澤良彰<sup>††</sup>

本発表では、パターンマッチ手法を用いて Java のバイトコードをあらかじめ用意した機械コードで実行することで、インタプリタ実行速度を高速化する手法を提案する。現在 Java プログラムの高速化の要は Just-in-time (JIT) コンパイラにより生成される機械コードによる実行であるが、JIT コンパイラによるコンパイル時間がかえってプログラムの実行を遅くしてしまう問題も知られている。この問題を解決する方法として、インタプリタ実行と JIT コンパイラによる機械コード実行を組み合わせた選択的コンパイル手法が提案されている。特にインタプリタ実行の高速化は、この選択的コンパイル手法を用いる際の性能向上に大きく影響を与える。我々は Java プログラム中に特定のパターンを持ったメソッドが多数出現し、実行されていることに着目した。これらのメソッドの多くは、Java 言語のオブジェクト思考的特性から、主にインスタンス変数やクラス変数へのアクセスのために記述されていた。本発表では、マスクしたワードパターンを用いたパターンマッチ手法により高速な検出処理を実現する手法を提案するとともに、パターンマッチのオーバーヘッドを考慮したうえでの本手法の有効性を示す。

### An Acceleration Technique of Java Interpreter Using the Bytecode Pattern-matching

TOSHIAKI YASUE,<sup>†</sup> AKITERU NAKA,<sup>††</sup> HIDEAKI KOMATSU<sup>†</sup>  
and YOSHIAKI FUKAZAWA<sup>††</sup>

In this paper, we propose an acceleration technique of Java interpreter by executing the pre-compiled codes that are selected using the bytecode pattern-matching of the method. While the Just-in-time (JIT) compiler boosts the execution speed significantly, its compilation time cause a serious problem in some programs. The selective compilation technique solves this compilation overhead problem to combine the execution of the interpreter and the native code compiled by JIT. Since the interpreting speed is much slower than the execution speed of the native code, the interpreting speed becomes important increasingly on the whole execution. We have marked that some kinds of the patterns of method's bytecodes are executed in many times through various Java programs, because most of them are written to access instance variables and class variables through the method as a typical object oriented programming style. This paper shows not only the implementation of the fast pattern-matching method using masked word patterns, but also the efficiency of our technique including the pattern-matching overhead through some evaluations.

(平成 12 年 6 月 16 日発表)

---

<sup>†</sup> 日本アイ・ビー・エム株式会社東京基礎研究所  
Tokyo Research Laboratory, IBM Japan

<sup>††</sup> 早稲田大学理工学部  
School of Science and Engineering, Waseda University