

発表概要

Ruby の文字列処理の手軽な高速化

中川 博貴^{†1} 笹田 耕一^{†1}

本発表では Ruby 処理系の正規表現などの文字列処理を手軽に高速化する手法について述べる。高速化を目指すにあたり、文字列処理のアルゴリズムに手を加えず、互換性を損なわない範囲で行うことができる 2 つの手軽な手法を適用した。1 つ目に、正規表現エンジン鬼車用の AOT コンパイラを実装した。Ruby の正規表現処理を行う正規表現エンジン鬼車は、仮想マシン型の評価器を持ち、実行時に正規表現をオPCODEに変換して評価する。我々は正規表現処理の高速化を目指し、このオPCODE列を C 言語の関数に変換し、仮想マシンの代わりに動作させる AOT コンパイラを開発した。2 つ目に、主要な文字コードについて、文字コードに関する処理を文字列処理へインライン化した。鬼車は多言語文字列に対応した正規表現エンジンである。多言語文字列に対応するために、文字のバイト数を調べるといった各文字コード固有の処理を関数ポインタによって呼び分けている。また、この仕組みを利用して、Ruby 処理系の文字列も多言語化が行われている。しかし、各文字コード固有の処理を関数ポインタ経由で呼び出すため、関数呼び出しにコストがかかり、また C コンパイラによる関数をまたいだ最適化が効きにくくなるという問題があった。そこで、我々は主要な文字コードについて、文字コードに関する処理を文字列操作にインライン化することで高速化を実現した。本発表では、鬼車用 AOT コンパイラの仕組みと文字コードに関する処理のインライン化の手法について述べる。そして評価を行い、これらの高速化の結果について述べる。

Lightweight Methods to Speed Up String Processing for Ruby

HIROKI NAKAGAWA^{†1} and KOICHI SASADA^{†1}

In this presentation, we describe the lightweight methods to speed up string processing for Ruby. To speed up, we adapt two lightweight methods which do not change string processing algorithms and do not break compatibility. First, we implemented the AOT compiler for regular expression engine Oniguruma which is used for regular expression processing in Ruby. Oniguruma is im-

plemented as a virtual machine. The regular expression parser of Oniguruma compiles the regular expressions into a sequence of opcodes, and then the virtual machine executes it. We implemented the AOT compiler which compiles an opcode sequence into a C function, and then executes the C function instead of the execution of virtual machine. Second, we inlined processes of major character encodings into string functions. Oniguruma supports multilingual strings. To support multilingual strings, Oniguruma calls each character encoding functions, for example a function to calculate a byte length of a character, via a function pointer. Similarly, Ruby string is multilingualized in the same way as Oniguruma. However this approach increases the cost of function calls, and prevents a C compiler interprocedural optimization. For speed up, we inlined processes of major character encodings into string functions. In this presentation, we describe the design and the implementation of the AOT compiler for Oniguruma and the method to inline processes of character encodings. Moreover, we show the results of performance evaluation for these implementations.

(平成 23 年 4 月 26 日発表)

^{†1} 東京大学大学院情報理工学系研究科

Graduate School of Information Science and Technology, The University of Tokyo