

発表概要

コールサイトローカルな型情報による 効率的な関数インライン展開

藤澤 徹^{1,a)}

2018年1月16日発表

動的型付き言語の高速化においては、型ごとに最適化したコードをいかに高速かつ効率的に生成するかが鍵となる。プロダクションコードの巨大化かつ抽象化にともないポリモーフィックな関数の最適化手法の重要度が増してきている。あるポリモーフィックな関数が、各コールサイトごとに見ればモノモーフィックとみなせる場合、コールサイトごとの型に特化した関数インライン展開を行うことで効率の良いコードが生成できることが広く知られている。しかし、従来から広く使用されている関数ごとの型フィードバックでは、関数をインライン展開する前の状態において複数のコールサイト由来の型情報を区別することができず、不要に汎用的なコードを生成してしまい、これを最適化するにはもう一段の JIT コンパイル層が必要となる。本発表では、この問題を改善する既存実装に適用可能な最適化手法として、オブジェクト構造を意識したシグネチャごとの型フィードバックを提案する。これにより、複数のコールサイト由来の型情報を区別可能となり最初のインライン展開においてコールサイトに特化した型情報を用いることができ、さらなる最適化を可能とする。本手法を Mozilla Firefox の JavaScript 処理系 SpiderMonkey に実装し、ポリモーフィックな関数のコールサイトごとの最適化への適用可能性を提示するとともに、各状況への最適化の可能性や性能向上、および手法の軽量さについて評価する。

Efficient Function Inlining with Callsite Local Type Information

TOORU FUJISAWA^{1,a)}

Presented: January 16, 2018

JIT compilers generate type-specialized code to improve performance of dynamically typed programming languages. As production code becomes huge and abstract, optimization of polymorphic functions gets increasingly important. If a polymorphic function can be considered monomorphic at each callsite, inline expansion specialized for types appeared in each callsite is known to generate efficient native code. Nevertheless, traditional compilation techniques with function-wise type feedback cannot distinguish types in different callsites, so unnecessarily generic code is often generated, which requires an extra JIT compiling tier to optimize inlined code. To address this issue, in this presentation, we propose a compilation technique with object-structure-aware per-signature type feedback which distinguishes type information from different callsites. This technique is applicable to existing implementation. Functions are inlined with type information specialized for the callsite in the first stage of JIT compilation, promoting further optimizations. We have implemented our method in Mozilla Firefox's JavaScript engine, SpiderMonkey. We discuss the applicability of our method for polymorphic functions. Our evaluation shows that, despite the simple implementation, the proposed method enhanced performance of benchmarks.

¹ 東京大学工学部電子情報工学科
Department of Electrical and Electronic Engineering, Department of Information and Communication Engineering, The University of Tokyo, Bunkyo, Tokyo 113-8656, Japan
^{a)} arai@eidous.ic.i.u-tokyo.ac.jp