

---

**発表概要**

---

## 効率的なプロファイリングを目的としたメソッドキャッシュにおける エントリ置換アルゴリズムの提案

長 田 忍<sup>†</sup> 檜 崎 修 二<sup>††</sup>

動的型付け OOP (Object Oriented Programming Language) では実行時に型付けを行うため、型情報を用いた静的解析が困難となる。したがって静的解析で得られた情報に基づくコンパイル時最適化を施すことができず、実行時における動的な最適化が処理系全体の高速化に必要となる。動的型付け OOP に適用できる最適化としてキャッシュ法があげられ、実装が容易であるメソッドキャッシュを利用するのが一般的である。メソッドキャッシュはメッセージ名 (セクタ) と引数の型の並びとを組としたシグネチャを鍵としてメソッドを検索するハッシュ表であり、そのヒット率は 90% 程度である。本研究では、最近よく使うメソッドがメソッドキャッシュ上にあるという性能を利用し、メソッドキャッシュにプロファイラとしての機能を持たせることを目的としている。具体的にはメソッドキャッシュ上のメソッド管理に使用するエントリにメソッドの呼び出しカウンタを持たせ、ヒットした場合は増加、ミスした場合は減少させることでプロファイリングを行う。これはエージングアルゴリズムと呼ばれる置換アルゴリズムを基にしたもので、メソッドキャッシュにおける効率的なエントリ置換アルゴリズムとして機能することからオーバーヘッドが少ないプロファイリングが可能となる。本発表では提案するエントリ置換アルゴリズムについて述べ、また動的型付け OOP である Squeak の仮想マシンへ提案手法を実装した効果について説明する。

### Proposal of an Entry Replacement Algorithm on Method-cache for Efficient Profiling

SHINOBU NAGATA<sup>†</sup> and SHUJI NARAZAKI<sup>††</sup>

In Object-Oriented Programming Languages (OOPs) whose interpreter determines types of variables in execution time, called dynamic typing OOPs, it is difficult to execute static analysis using type information. For this reason, total speedup of interpreters needs runtime dynamic optimizations. One of them is known as *method-cache* uses a hash table maps a pair of a message selector and signature to a method body that should be invoked, and it is used in many dynamic typing OOPs widely. In this research, we claim that method-cache can be a method profiler because it is sure to store methods used frequently. In view of this, our method-cache has a counter at each entry, and it increments the counter if a method is hit. If a method is not hit, the method-cache decrements the counter and decides which entry should be replaced. This algorithm is based on the replacement algorithm named *aging algorithm*, and makes method-cache a simple, efficient profiler because the proposed entry replacement algorithm optimizes contents of method-cache. In this presentation, we explain the proposed entry replacement algorithm. Furthermore, we describe its implementation and evaluations on virtual machine of Squeak.

(平成 19 年 1 月 19 日発表)

---

<sup>†</sup> 長崎大学大学院生産科学研究科

Graduate School of Science and Technology, Nagasaki  
University

<sup>††</sup> 長崎大学工学部情報システム工学科

Department of Computer and Information Sciences,  
Faculty of Engineering, Nagasaki University