Multicodes: Characterization and Optimization of Java Applications
Ben Stephenson and Wade Holst, University of Western Ontario

Abstract

Languages that use a virtual machine (VM), like Java, provide an abstract API with a substantial performance penalty compared to languages that do not use a VM. In this paper, we focus on the Java Virtual Machine (JVM) and determine if performance gains can be achieved using multicodes because much of the work can be performed ahead of time. This will remove the need to preserve a record of each method invoked for each program, as is required with despecialization and/or instruction widening. Timing results were also performed for the top multicodes determined using this algorithm. However, due to limitations in the JVM, not all classes can be modified to use multicodes.

Introduction

A multicode is a library of bytecode subroutines that are pre-compiled into the JVM. The JVM and other virtual machines typically use two-arity operation sequences. In languages that use a virtual machine, the performance penalty is much higher than with other languages. To reduce this penalty, multicodes have been introduced in the past to allow for a single bytecode sequence to be replaced with a subroutine that contains multiple bytecode sequences.

Related Readings


Multicodes: Characterization and Optimization of Java Applications
Ben Stephenson and Wade Holst, University of Western Ontario

Abstract

Languages that use a virtual machine (VM), like Java, provide an abstract API with a substantial performance penalty compared to languages that do not use a VM. In this paper, we focus on the Java Virtual Machine (JVM) and determine if performance gains can be achieved using multicodes because much of the work can be performed ahead of time. This will remove the need to preserve a record of each method invoked for each program, as is required with despecialization and/or instruction widening. Timing results were also performed for the top multicodes determined using this algorithm. However, due to limitations in the JVM, not all classes can be modified to use multicodes.

Introduction

A multicode is a library of bytecode subroutines that are pre-compiled into the JVM. The JVM and other virtual machines typically use two-arity operation sequences. In languages that use a virtual machine, the performance penalty is much higher than with other languages. To reduce this penalty, multicodes have been introduced in the past to allow for a single bytecode sequence to be replaced with a subroutine that contains multiple bytecode sequences.

Related Readings
