Adaptive Inlining and On-Stack Replacement in the CACAO Virtual Machine

Institut für Computersprachen
Technische Universität Wien
Austria

Edwin Steiner    Andreas Krall    Christian Thalinger
Overview

1. Introduction

2. Adaptive Optimization Framework
   - Modules of the Framework
   - Adaptive Recompilation
   - Inlining

3. On-Stack Replacement
   - Execution/Source State
   - Replacement Points

4. Empirical Evaluation

5. Conclusion and Further Work
Introduction

- object oriented programming style
- optimize program by method inlining
- virtual (synchronized) methods, exceptions
- dynamic class loading (undo)
- adaptive optimization
- CACAO - JIT for multiple architectures
Adaptive Optimization Framework

Modules of the Framework

- inliner
  - inline

- compiler
  - jit-compile

- garbage collector
  - suspend-self

- code repository
  - invalidate-code
  - get-current-code
  - request-optimization

- replacement
  - activate-rp-points
  - gc-read-source-state
  - gc-write-source-state
  - gc-set-traps
  - replace-me

- arch. layer
  - replace-in
  - patch-rp-point
  - replace-out

- method db
  - record-assumption
  - break-assumption

- linker

- compiled code
  - replacement point
Adaptive Recompilation

- baseline compiler (countdown traps)
- recompilation with instrumentation
- recompilation with optimizations
- deoptimization when assumptions become invalid
- on-stack replacement
Inlining

- inlining works on intermediate representation
- profile guided (caller of hot methods)
- inlining heuristics
  - aggressive depth-first
  - aggressive breadth-first
  - knapsack
On-Stack Replacement

- execution state
  - snapshot of registers and machine stack
- source state
  - values of Java variables and Java operand stack
- replacement points
  - allocation of data
  - traps
Empirical Evaluation

Execution Times and Number of Executed Calls

Andreas Krall (TU Wien)
Empirical Evaluation

Code Size Changes (Knapsack Heuristics)

Andreas Krall (TU Wien)  Adaptive Inlining and On-Stack Replacement  September 7, 2007  9 / 10
Conclusion

- adaptive compilation framework for CACAO
- different inlining heuristics evaluated
- up to 99% of calls eliminated
- up to 18% speedup
- further improvements by linear scan register allocator expected
- www.cacaojvm.org