Type Feedback for Bytecode Interpreters

Position Paper, ICOOOLPS 2007

Michael Haupt, HPI Potsdam
Robert Hirschfeld, HPI Potsdam
Marcus Denker, SCG Bern
Message Lookup

Method call in dynamic object oriented languages:

1. Message is send to an object
2. The class of the object is searched for a method
3. If not found, go to superclass
4. Execute the method found or raise exception

Problem: slow
Global Cache

- `<target class, selector> → Method`
- Frequent collisions
- Flushed often
  - Any change in the class hierarchy
  - GC
- No information *per send site*
Inline Caching

- Cache per send site \( \text{class} \rightarrow \text{method} \)
  - Simple Inline Cache: one entry
  - PIC: multiple entries

- Provides type-feedback data
- Only used in VMs with JIT compilers
  - Examples: Self, Strongtalk, some Java VMs
Interpreter PIC

- Why no PIC in Interpreters?
- Performance?
- Type feedback data!
Prototype

- Squeak Smalltalk
- Very simple interpreter
- Easy to modify
- Uses global lookup cache
PICs as Objects

• PICs should be objects, not VM level data structures

• VM level modification minimized

• All management code implemented in high-level language

• PIC data reflectively accessible
Implementation

Method with Inline Cache

PIC

(CompiledMethod instance literal slot)

polymorphic inline cache object
Status

• Image side code for PIC implemented
  • Compiler / method layout modifications
• First virtual machine experiments

• Future: use dynamic feedback for optimizations (AOStA)
Conclusion

- Experiment with PICs in the Interpreter
- PICs as Objects
- Implement as much as possible outside the VM
Conclusion

• Experiment with PICs in the Interpreter
• PICs as Objects
• Implement as much as possible outside the VM

Feedback?