A bytecode set for adaptive optimizations

Clément Béra & Eliot Miranda
Introduction

• The Cog VM is the standard VM for:
  • Pharo
  • Squeak
  • Newspeak
Introduction

• Working runtime bytecode to bytecode optimizer for Cog’s JIT

• The optimizer
  • depends the bytecode set quality
  • needs new bytecode instructions
Introduction

• Design of a new bytecode set
Plan

- Context
- Challenges for a good bytecode set
- Current Issues
- New bytecode set
- Switching between bytecode sets
Context

- Compiled methods are objects
- Shared between the VM and the image
Memory representation of Compiled Method in 32 bits with the new Memory Manager Spur

- Object header (8 bytes)
- compiled method header (4 bytes)
- literals (4 bytes per literal)
- bytecodes (variable)
- source pointer (variable, usually 4 bytes)
Memory representation of Compiled Method in 32 bits with the new Memory Manager Spur

Object header (8 bytes)

compiled method header (4 bytes)

literals (4 bytes per literal)

bytecodes (variable)

source pointer (variable, usually 4 bytes)
Bytecode set

- Stack based
- Interpreted by the StackInterpreter
- Compiled to machine code by Cogit
- Generated by the in-image compiler
Challenges

- Generic challenges

- Challenges for the bytecode optimizer
Generic challenges

- Platform-independent
- Compact
- Easy to decode
- Backward compatibility
Optimizer challenges

...
Optimizer

Thursday there’s a talk about it.

- Inlining
- Primitive specialization
Optimizer challenges

- Inlined primitives / unsafe operations
- Large methods
- Access to non receiver instance variable
- Extendable
Current issues

- Large method unsupported
- Few available bytes
- Primitive encoding forbidding inlining
- DoubleExtendedDoAnything bytecode
- Immediate objects compaction
- Late addition of the closure bytecode
New bytecode set

• Extendable instructions (prefix)
• Inlined primitives / unsafe operations
• Extendable (available bytes)
• Maximum number of literals increased
New bytecode set

• Overall bytecode size smaller
• Immediate objects compaction
• Easier decoding
  • Sorted by number of bytes
  • Sorted by functionalities
• Closure decoding improved
Bytecode set switch

- Offline converter: hard to debug
- VM supporting two bytecode set
- Bit in compiled method header
Conclusion

- Designed a bytecode set for runtime bytecode to bytecode optimizations
  
- Next step is to work on the optimizer
  
- Come at my talk Thursday about it