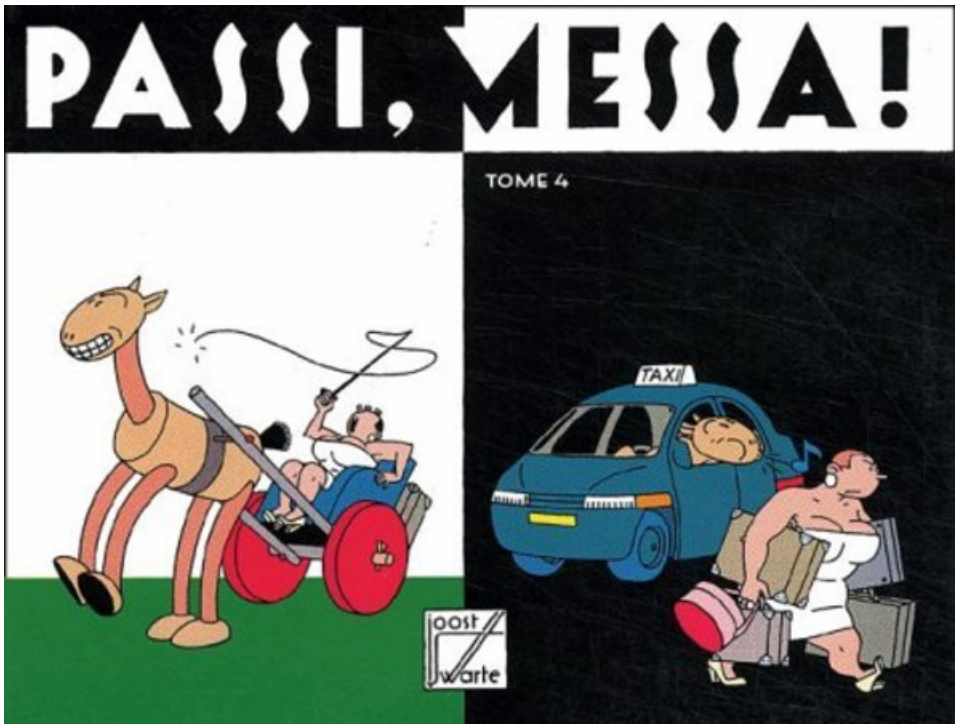
Soops Group
Amsterdam
Würzburg

Code Optimization



Adriaan van Os
ESUG 2006

Soops Group
Amsterdam
Würzburg

Introduction

- Working at Soops since 1995
- Customers include
  - Research (Economical)
  - Exchanges (Power, Gas)
  - Insurance
- Projects in
  - VisualWorks
  - GemStone
  - VisualAge

Soops Group
Amsterdam
Würzburg

Demanding projects

- Data-intensive
  - Rule based data warehouse like application built with VisualWorks and GemStone.
- Calculation-intensive
  - Decentralized coupling of electricity markets done with VisualWorks

Soops Group
Amsterdam
Würzburg

Prologue

Code Optimization

Adriaan van Os
ESUG 2006

Soops Group
Amsterdam
Würzburg

Controversial

- Hard to predict what piece of code might become a problem
- Often there won't be a problem
- Optimizations will break some (Smalllint) rules
- Optimized systems can become harder to extend

Soops Group
Amsterdam
Würzburg

Context

- Optimizations are often not reusable
- Any change might outdate them
  - VM
  - Image
  - Platform
- Demand for UnitTests

Soops Group
Amsterdam
Würzburg

Strategy Against Performance Problems

- Concentrate on the design first
- Result too slow?
  - Analyze it
    - Tools
  - Solve it
    - Tips
    - Tricks
  - Test it
    - Tests

Soops Group
Amsterdam
Würzburg

Tools

Code Optimization

Adriaan van Os
ESUG 2006

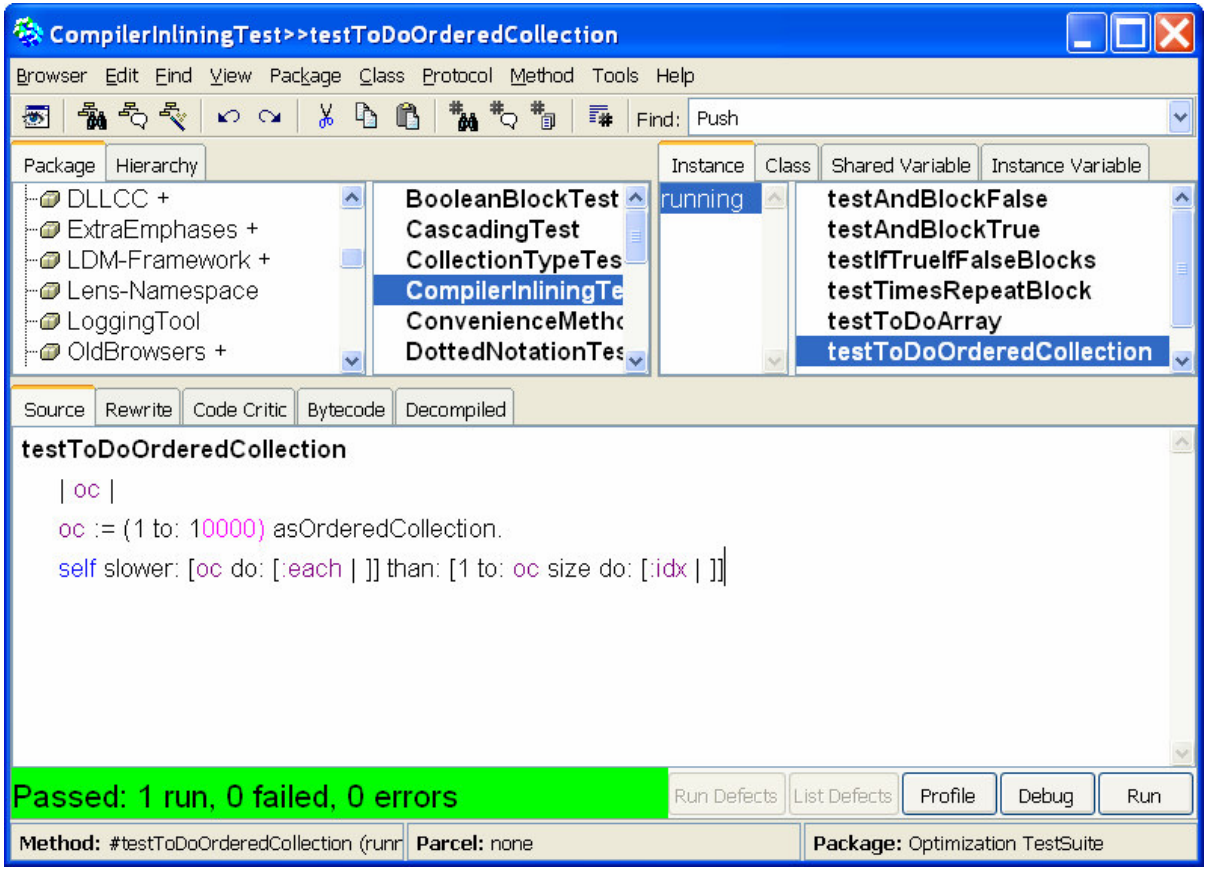
Soops Group
Amsterdam
Würzburg

Tools

- For analyses
  - Time millisecondsToRun: []
  - (Multi)TimeProfiler (VW)
  - (Multi)AllocationProfiler (VW)
- For inspiration
  - A few Smalllint rules
  - RBByteCodeTool (VW)
  - RBDecompiledTool (VW)

Soops Group
Amsterdam
Würzburg

VW RB Integration



Soops Group
Amsterdam
Würzburg

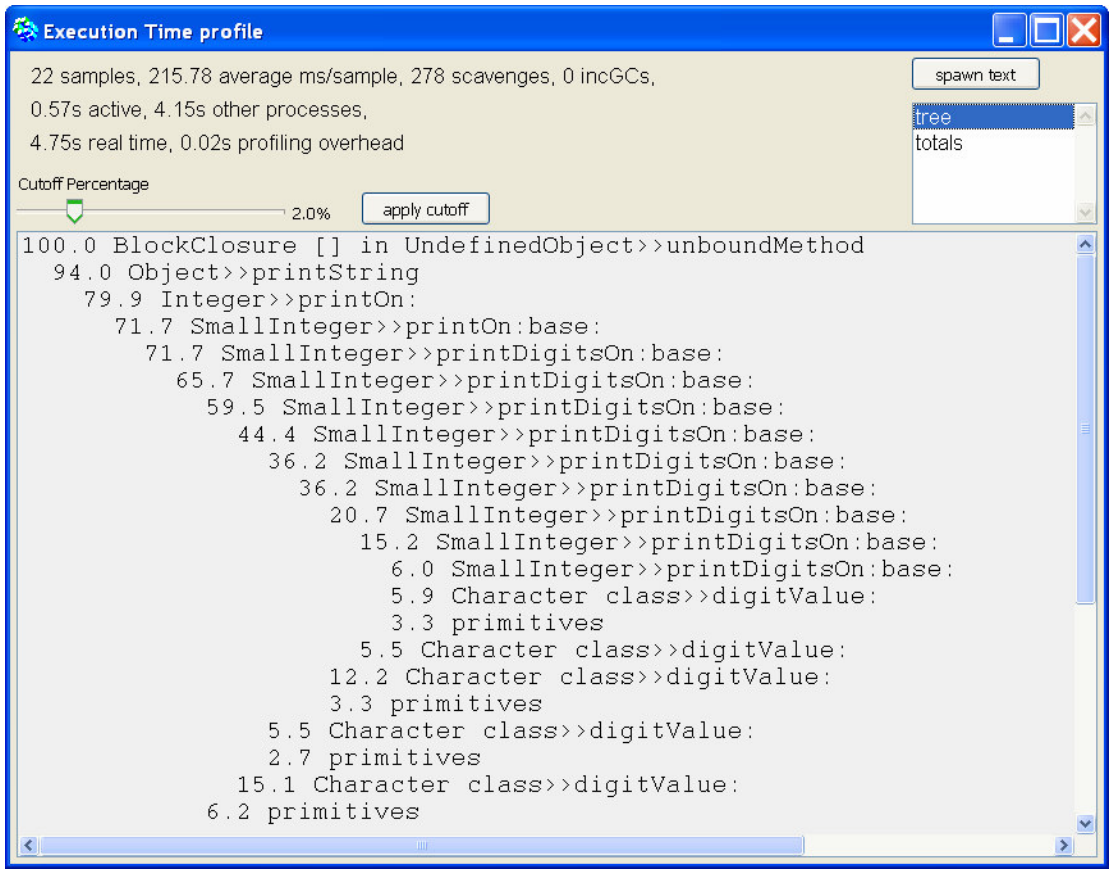
Time millisecondsToRun: []

- Run multiple times
  - 10000 timesRepeat: []
- Beware of large integers
- Beware of allocation/garbage collector
- Sometimes it's still hard to get consistent results ☹

Soops Group
Amsterdam
Würzburg

TimeProfiler

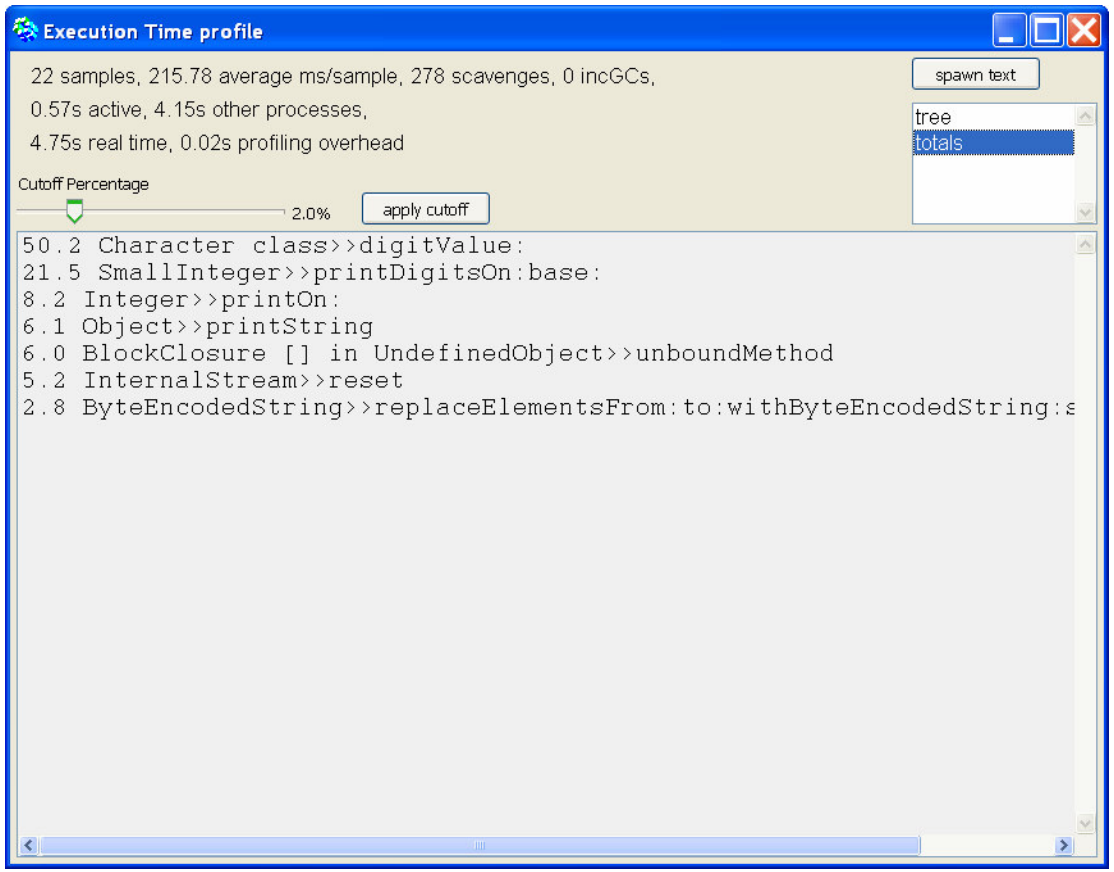
TimeProfiler profile: [1000000 timesRepeat:[123456789 printString size]]



Soops Group
Amsterdam
Würzburg

TimeProfiler

TimeProfiler profile: [1000000 timesRepeat:[123456789 printString size]]



Soops Group
Amsterdam
Würzburg

Tips & Tricks

Code Optimization

Adriaan van Os
ESUG 2006

Soops Group
Amsterdam
Würzburg

Selectors

- Special selectors
  - Specialized opcode will be generated if syntactic requirements are met
  - Can't be overwritten
  - See DefineOpcodePool class
  - The set of selectors can be modified
- Optimized selectors
  - Transformed if syntactic requirements are met
  - Selectors will be inlined
  - Can't be overwritten
  - See MessageNode class
  - You can add your own transformations
- Primitives
  - Write your own

Soops Group
Amsterdam
Würzburg

Selector Transformations

transformIfNil

```
"MacroSelectors at: #ifNil: put: #transformIfNil“

 ^((self testLiteralBlock: 0 at: 1)
            and: [self receiver hasEffect not])
        ifTrue:
            [ConditionalNode new
                sourcePosition: self sourcePosition;
                condition: (MessageNode new
                    receiver: (LiteralNode new value: nil)
                    selector: #==
                    arguments: (Array with: self receiver))
                trueBlock: arguments first
                falseBlock: (BlockNode new body: self receiver)
                from: self]

        ifFalse: [nil]
```

Soops Group
Amsterdam
Würzburg

Special and Optimized Selector Examples

- Use and: instead of && even if you have the argument ready
- The VW compiler warns you for aBoolean and: aBlock …
- …but aBoolean and: [aBlock value] doesn't seem to be faster
- It's true for, eg. timesRepeat:
- Compiler inlines self do: in SequenceableCollection, but not outside

Soops Group
Amsterdam
Würzburg

Inlining

- Get rid of those message sends
- Inline cases the compiler don't know about

Soops Group
Amsterdam
Würzburg

Blocks

- Keep them clean
  - Declare variables in innermost scope
  - Avoid assigning values to outer scope variables
  - Avoid return (^) inside the block
- Some clean blocks can be inlined
  - No instance of BlockClosure is created
  - Share context with sender

Knights of the Square Bracket

Soops Group
Amsterdam
Würzburg

Numbers

- Avoid coercion
- Avoid LargeIntegers
- Avoid Fractions
  - Avoid sending / and // with Integers
  - Finding gcd's is very expensive
- Use Doubles
- Higher generality first
  - 10.0 * 10

Soops Group
Amsterdam
Würzburg

Collections and Iterations

- Avoid intermediates and repeated iterations
  - Use aDict keysDo:, not aDict keys do:
    - Avoid keys anyway
  - Implement
    - select:do:
    - select:collect:
    - collect:select:
  - Use modify:, not collect: if possible
- Pregrow collections

Soops Group
Amsterdam
Würzburg

Collections and Inlining

- Enable inlining
  - anArray do: [:each|]
    - 1 to: anArray size do: [:idx|]
      - Not (1 to: anArray size) do: [:idx|]
  - anOrderedCollection firstIndex to: lastIndex do: [:|idx|]
    - Used to be faster than

      1 to: anOrderedCollection size do: [:idx|]
    - Slower in VW 7.4.1

Soops Group
Amsterdam
Würzburg

Collection Types

- For faster lookups, use
  - Set/Dictionary
  - IdentitySet/IdentityDictionary
    - Beware of maximum identity hash
    - Implement your own hash algorithm
  - RBSmallDictionary
    - For very small collections
- For faster iterations, use
  - Array
- For faster growing, use
  - OrderedCollection
- Hybrid
  - OrderedDictionary

Soops Group
Amsterdam
Würzburg

Unnecessary Code

- aDictionary values
- aCollection asSortedCollection first
  - Use aCollection fold: [:a :b | a min: b]
- aCollection asOrderedCollection first
  - Implement any
- aColllection contains: [each| each asUppercase = target asUppercase]
  - Keep static code out of the block
- aCollection reverse do:
  - Use reverseDo:

Soops Group
Amsterdam
Würzburg

Conditionals

- If you have a lot of conditionals you maybe have too few classes
- Common cases first

Soops Group
Amsterdam
Würzburg

Caching

- Don't do anything twice
- Make cache lookups fast
- Keep cache management simple

Soops Group
Amsterdam
Würzburg

GemStone

- Just a dialect(?)
  - All previous slides apply
- Objects on disk
  - Makes usage of identity more preferable
- Objects on other side of the wire
  - Reducing round trips is a major design issue
  - Minimize copying (replicating)
- Shared Objects
  - Garbage collection is harder
- Use the specialized collection types

Soops Group
Amsterdam
Würzburg

Tests

Code Optimization

Adriaan van Os
ESUG 2006

Soops Group
Amsterdam
Würzburg

Test Example

```
testToDoOrderedCollection
    | oc |
    oc := (1 to: 10000) asOrderedCollection.
    self slower: [oc do: [:each | ]] than: [1 to: oc size do: [:idx | ]]
```

Soops Group
Amsterdam
Würzburg

Test Method

```
slower: aSlowBlock than: aFastBlock
    | slowCount slowTime fastCount fastTime faktor |
    slowCount := fastCount := 1.

    ObjectMemory garbageCollect.

    [(slowTime := TimemillisecondsToRun: [slowCount timesRepeat: [aSlowBlock
    value]]) < 200] whileTrue: [slowCount := slowCount * 2].

    ObjectMemory garbageCollect.

    [(fastTime := Time millisecondsToRun: [fastCount timesRepeat: [aFastBlock value]])
    < 200] whileTrue: [fastCount := fastCount * 2].

    faktor := slowTime / slowCount / (fastTime / fastCount).

    self report: aSlowBlock slowerThan: aFastBlock factor: faktor.
    self assert: faktor > 1
            description: aSlowBlock method decompiledSource , ' is slower than '
                                    , aFastBlock method decompiledSource
```

Soops Group
Amsterdam
Würzburg

Conclusions

Code Optimization

Adriaan van Os
ESUG 2006

Soops Group
Amsterdam
Würzburg

Conclusions

- Concentrate on design first
- Don't try to predict problems
- Easy wins with
  - Inlining Blocks
  - Caching
  - Choosing the right collection types
- Test your tricks
- Test again with new VM, image, hardware, etc

Soops Group
Amsterdam
Würzburg

References

- Efficient Smalltalk
  - Travis Griggs, Smalltalk Solutions 2006
- VisualWorks Implementations Limits (PDF)
  - Cincom
- VisualWork Optimization (PDF)
  - Bernard Horan, Laura Hill, Mario Wolezko
- The Hitch Hiker's Guide to the Smalltalk Compiler
  - Vassili Bykov
- Niet zo, maar zo
  - Adriaan van Os

Soops Group
Amsterdam
Würzburg

PRAGUE
esug'06

Questions?

Thanks

adriaan @ soops . nl