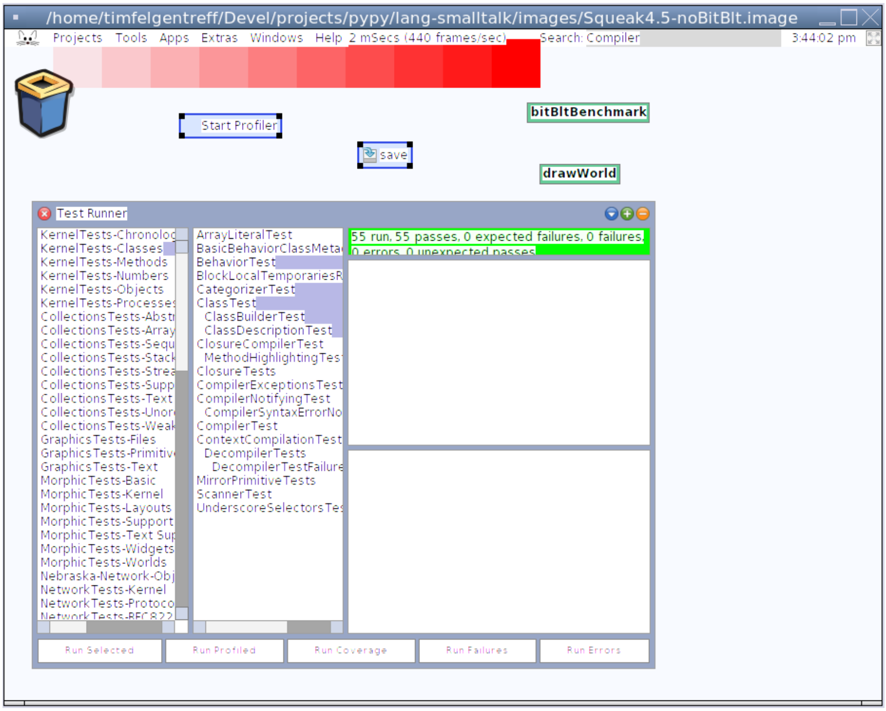
Spur to go faster:
Low-level Functionality in a High-Level Language

Master Project Proposal, SS 2015
Software Architecture Group, Prof. Dr. Robert Hirschfeld

R/SqueakVM

The R/SqueakVM is a virtual machine for Squeak/Smalltalk written in RPython and developed at our group. It is based on the RPython translation toolchain that also powers the fast PyPy interpreter, and re-uses its virtual machine translation mechanism and garbage collection implementation. The R/SqueakVM implements the standard Squeak bytecode set and combines powerful techniques such as storage strategies and shadow objects with a meta-tracing JIT to achieve performance that makes it feasible to run many of the functions that are traditionally implemented as part of the VM from pure Smalltalk. This includes primitives for handling large integers and composing the display, and thus allows us to experiment with and extend such core functionality from with the Squeak environment itself.



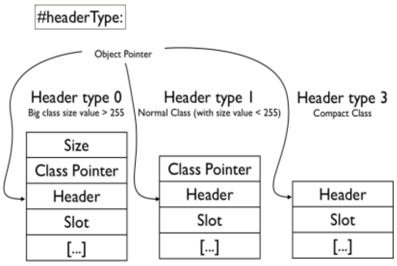
```
KernelTests-Chronolog
KernelTests-Classes
KernelTests-Methods
KernelTests-Numbers
KernelTests-Objects
KernelTests-Processes
CollectionsTests-Abstr
CollectionsTests-Array
CollectionsTests-Sequ
CollectionsTests-Stack
CollectionsTests-Strea
CollectionsTests-Supp
CollectionsTests-Text
CollectionsTests-Unor
CollectionsTests-Weak
GraphicsTests-Files
GraphicsTests-Primitiv
GraphicsTests-Text
MorphicTests-Basic
MorphicTests-Kernel
MorphicTests-Layouts
MorphicTests-Support
MorphicTests-Text Sup
MorphicTests-Widgets
MorphicTests-Worlds
Nebraska-Network-Obj
NetworkTests-Kernel
NetworkTests-Protoco
NetworkTests-RFC822
```

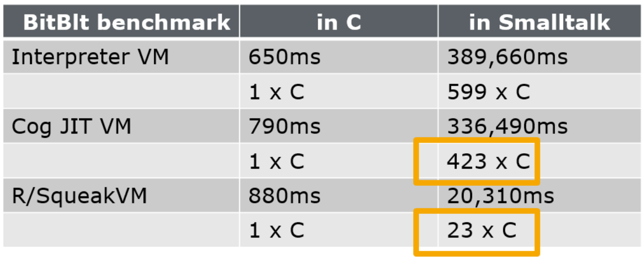
```
ArrayLiteralTest
BasicBehaviorClassMeta
BehaviorTest
BlockLocalTemporariesR
CategorizerTest
ClassTest
  ClassBuilderTest
  ClassDescriptionTest
ClosureCompilerTest
  MethodHighlightingTes
ClosureTests
CompilerExceptionsTest
CompilerNotifyingTest
  CompilerSyntaxErrorNo
CompilerTest
ContextCompilationTest
  DecompilerTests
    DecompilerTestFailure
MirrorPrimitiveTests
ScannerTest
UnderscoreSelectorsTes
```

```
55 run, 55 passes, 0 expected failures, 0 failures,
0 errors, 0 unexpected passes
```

Project Objectives

The R/SqueakVM in its current state is a performant way to run Squeak, but not to work in it. We currently have no way of saving a Squeak image. Additionally, recent benchmarks suggest that the image format has a negative impact on garbage collector performance. A new image format for Squeak, SPUR, has been proposed to alleviate this latter problem by greatly simplifying the object and memory layout of the Squeak image.





BitBlt benchmark	in C	in Smalltalk
Interpreter VM	650ms	389,660ms
	1 x C	599 x C
Cog JIT VM	790ms	336,490ms
	1 x C	423 x C
R/SqueakVM	880ms	20,310ms
	1 x C	23 x C

The goal of this project will be to implement reading and storing SPUR format images for R/SqueakVM. As a continuation of the work done previously, as much as possible of the functionality should be implemented in pure Smalltalk. If the only reason not to do so is performance, instead of implementing the functionality in a low-level language, your goal is to improve and extend the JIT techniques used in the VM to make it fast. In the end, the goal is to have a package with Smalltalk code that, when filed into an image, allows that image to run on R/SqueakVM with most of its functionality running in pure Smalltalk.

Contact

Prof. Dr. Robert Hirschfeld, Tim Felgentreff, Tobias Pape, Marcel Taeumel
Software Architecture Group, Potsdam
http://www.hpi.uni-potsdam.de/swa, hirschfeld@hpi.uni-potsdam.de