

Introducing Topaz

A Ruby built with the RPython toolchain

Topaz?

Ruby interpreter



Topaz?

Ruby interpreter

Written in

RPython

Topaz?

Ruby interpreter

written in
RPython

Topaz?

translated to C
and compiled

Ruby interpreter

written in
RPython

Topaz?

with a generated
JIT

translated to C
and compiled

```
(1..10_000).each do |i|  
  i <=> 5  
end
```

Topaz?

written in
RPython

with a generated
JIT

translated to C
and compiled

```
(1..10_000).each do |i|  
  i <=> 5  
end
```

```
@classdef.method("<=>")  
def method_comparator(self, space, w_o  
  if space.is_kind_of(w_other, space  
    other = space.float_w(w_other)  
    if self.floatvalue < other:  
      return space.newint(-1)  
    elif self.floatvalue == other:  
      return space.newint(0)  
    elif self.floatvalue > other:
```

Topaz?

with a generated
JIT

translated to C
and compiled


```
(1..10_000).each do |i|  
  i <=> 5  
end
```

```
@classdef.method("<=>")  
def method_comparator(self, space, w_o  
  if space.is_kind_of(w_other, space  
    other = space.float_w(w_other)  
    if self.floatvalue < other:  
      return space.newint(-1)  
    elif self.floatvalue == other:  
      return space.newint(0)  
    elif self.floatvalue > other:
```

Topaz?

with a generated
JIT

```
uct pypy_topaz_objects_objectobject_W.RootObject0 *pypy_g_W.FloatObject_method_comparator(struct  
double l_other 18; void* l_result 1818; void* l_result 1819;  
void* l_result 1820; void* l_result 1821; Signed l_totalsize 976;  
Signed l_totalsize 977; Signed l_totalsize 978;  
Signed l_totalsize 979; Signed l_v260949; Signed l_v260951;  
Signed l_v260953; Signed l_v260954; Signed l_v260964;  
Signed l_v260973; Signed l_v261005; Signed l_v261037;  
Signed l_v261069; bool_t l_v260947; bool_t l_v260958;  
bool_t l_v260967; bool_t l_v260969; bool_t l_v260971;  
bool_t l_v260972; bool_t l_v260974; bool_t l_v260980;  
bool_t l_v260981; bool_t l_v260986; bool_t l_v260993;  
bool_t l_v260997; bool_t l_v261002; bool_t l_v261004;  
bool_t l_v261006; bool_t l_v261012; bool_t l_v261013;  
bool_t l_v261016; bool_t l_v261025; bool_t l_v261029;  
bool_t l_v261034; bool_t l_v261036; bool_t l_v261038;  
bool_t l_v261044; bool_t l_v261045; bool_t l_v261050;  
bool_t l_v261057; bool_t l_v261061; bool_t l_v261066;  
bool_t l_v261068; bool_t l_v261070; bool_t l_v261076;  
bool_t l_v261077; bool_t l_v261082; bool_t l_v261089;  
bool_t l_v261093; bool_t l_v261098; bool_t l_v261116; char l_v260962;  
double l_v260965; double l_v260966; double l_v260968;  
double l_v260970; double l_v261101; double l_v261109;  
struct pypy_headers *l_v260983; struct pypy_headers *l_v261015;  
struct pypy_headers *l_v261047; struct pypy_headers *l_v261079;  
struct pypy_objects *l_v260959; struct pypy_objects *l_v260987;  
struct pypy_objects *l_v261019; struct pypy_objects *l_v261051;  
struct pypy_objects *l_v261083;  
struct pypy_object_vtable0 *l_v260957;
```

```
(1.1..10_000).each do |i|
  i <=> 5
end
```

```
@classdef.method("<=>")
def method_comparator(self, space, w_o
  if space.is_kind_of(w_other, space
    other = space.float_w(w_other)
    if self.floatvalue < other:
      return space.newint(-1)
    elif self.floatvalue == other:
      return space.newint(0)
    elif self.floatvalue > other:
```

Topaz?

```
getfield_gc_l_pure %r0, FieldDescr(<GcSt
-live- %f0, %r1
guard_class %r1 -> %i0
getfield_raw_i_pure %i0, FieldDescr(<Str
-live- %f0, %i0, %r1
int_guard_value %i0
residual_call_irf_f %i0, <IndirectCallTar
-live- %f0, %f1
float_lt %f0, %f1 -> %i0
goto_if_not_int_is_true %i0, L2
-live- L2
ref_return $<* struct topaz.objects.object
---
L2:
ref_return $<* struct topaz.objects.object
```

ated

```
uct pypy_topaz_objects_objectobject_W.RootObject0 *pypy_g_W_FloatObject_method_comparator(struct
double l_other 18; void* l_result 1818; void* l_result 1819;
void* l_result 1820; void* l_result 1821; Signed l_totalsize 976;
Signed l_totalsize 977; Signed l_totalsize 978;
Signed l_totalsize 979; Signed l_v260949; Signed l_v260951;
Signed l_v260953; Signed l_v260954; Signed l_v260964;
Signed l_v260973; Signed l_v261005; Signed l_v261037;
Signed l_v261069; bool_t l_v260947; bool_t l_v260958;
bool_t l_v260987; bool_t l_v260969; bool_t l_v260971;
bool_t l_v260972; bool_t l_v260974; bool_t l_v260980;
bool_t l_v260981; bool_t l_v260986; bool_t l_v260993;
bool_t l_v260997; bool_t l_v261002; bool_t l_v261004;
bool_t l_v261006; bool_t l_v261012; bool_t l_v261013;
bool_t l_v261016; bool_t l_v261025; bool_t l_v261029;
bool_t l_v261034; bool_t l_v261036; bool_t l_v261038;
bool_t l_v261044; bool_t l_v261045; bool_t l_v261050;
bool_t l_v261057; bool_t l_v261061; bool_t l_v261066;
bool_t l_v261068; bool_t l_v261070; bool_t l_v261076;
bool_t l_v261077; bool_t l_v261082; bool_t l_v261089;
bool_t l_v261093; bool_t l_v261098; bool_t l_v261116; char l_v260962;
double l_v260965; double l_v260966; double l_v260968;
double l_v260970; double l_v261101; double l_v261109;
struct pypy_headers *l_v260983; struct pypy_headers *l_v261015;
struct pypy_headers *l_v261047; struct pypy_headers *l_v261079;
struct pypy_objects *l_v260959; struct pypy_objects *l_v260987;
struct pypy_objects *l_v261019; struct pypy_objects *l_v261051;
struct pypy_objects *l_v261083;
struct pypy_object_vtable0 *l_v260957;
```

Topaz is not
a VM that runs
Python AND Ruby

Ruby interpreter
written in RPython
Topaz?
with a generated JIT
translated to C and compiled

Why?

<_ko1> Alex_Gaynor: can i ask the motivation of this project? hobby?

<Alex_Gaynor> _ko1: for fun, to demonstrate that rpython is a good platform for all kinds of VMs

11 months ago

me



@cfbolz



@alex_gaynor

What is RPython?

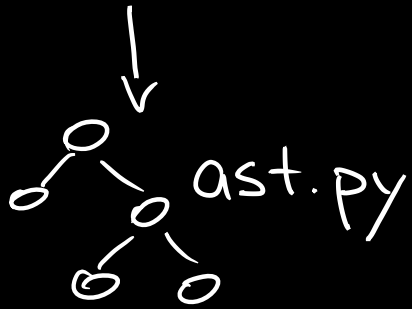
RPython is a restricted subset of Python that is amenable to static analysis

The property of “being RPython” always applies to a full program

The exact definition is “RPython is everything that our translation toolchain can accept” :)

RPython is a language to write interpreters in

lexer.py
parser.py

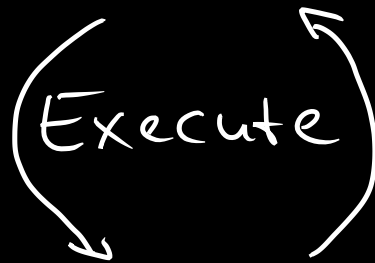


Bytecode

→ consts.py



Start here
Object Space



interpreter.py

Core Classes
w-Object

Core Ruby Code
/lib-topaz

Current
Status

≈ 5000 passing RubySpec expectations

missing almost all stdlib, and
many methods

Thanks Rubyspec
contributors! ✓

You're awesome! ✓

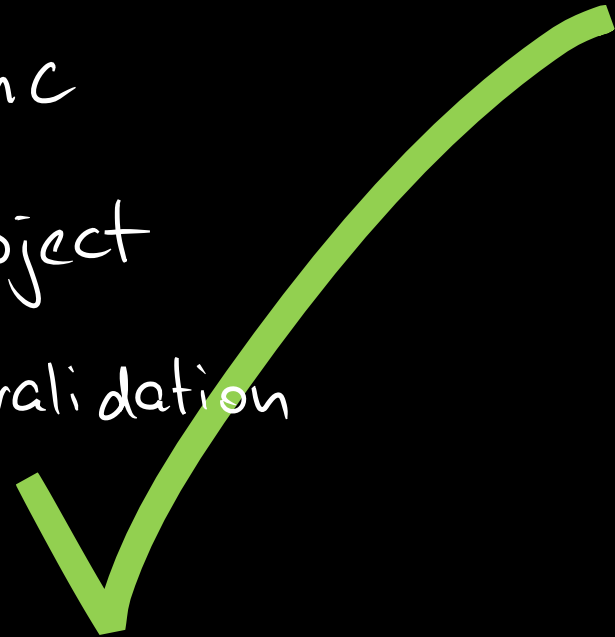
"Hard Problems" (acc to @headius & @evanphx)

Full Binding#eval

Kernel #set_trace_func

ObjectSpace.each_object

Constant lookup invalidation



"Hard Problems" (acc to @headius & @evanphx)

Encoding support

Threads

Fibers

FFI



≈ 15,000 lines Python

≈ 2,500 lines Ruby

Very promising benchmarks

(... but few run)

We need your help

What was that bit
about performance?

Takeaways (free!)

We want to be
the fastest

We are going to
be the fastest

(eventually)

Come and join
us on the way

#topaz.freenode.net
@topazproject
git:topazproject/topaz
http://topazruby.com

@timfelgentreff
#phlebas