Thread-Safe Built-in Collections for Dynamic Languages

Benoit Daloze, Arie Tal, Stefan Marr, Hanspeter Mössenböck, Erez Petrank

Thread-Safety Problems

```ruby
array = []
# Create 100 threads
100.times.map {
  Thread.new {
    # Append 1000 integers to the array
    1000.times { |i|
      array << i
    }
  }
}.each { |thread| thread.join }
puts array.size
```

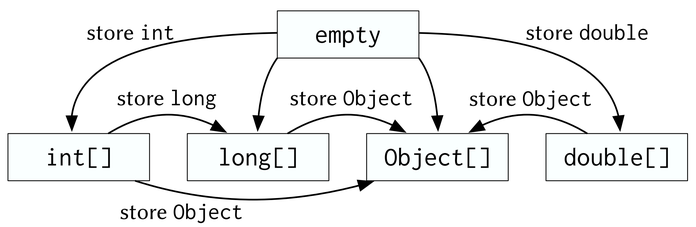
MRI, with a GIL:

```
ruby append.rb
100000
```

JRuby, with concurrent threads:

```
jruby append.rb
ConcurrencyError: Detected invalid array
contents due to unsynchronized modifications
    << at org/jruby/RubyArray.java:1256
```

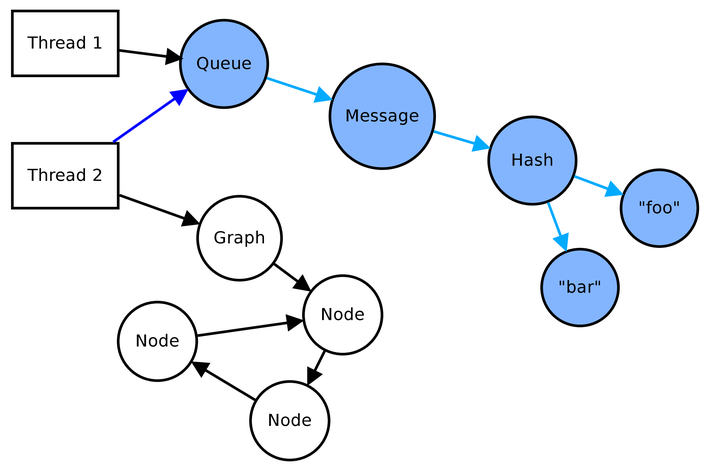
Storage Strategies




```ruby
array = []       # empty
array << 1       # int[]
array << 2**42   # long[]
array[0] = "Hi"  # Object[]

Array.new(42, 0.0) # double[]
```

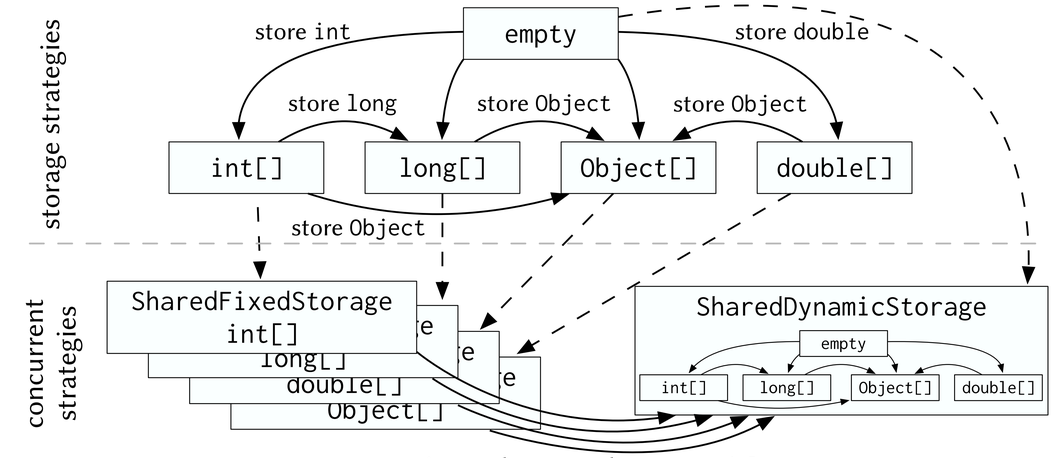
Storage Strategies for Collections in Dynamically Typed Languages, Bolz, Diekmann and Tratt, OOPSLA'13

Tracking sharing of collections



Efficient and Thread-Safe Objects for Dynamically-Typed Languages, Daloze, Marr, Bonetta and Mössenböck, OOPSLA'16

Idea: Concurrent Strategies



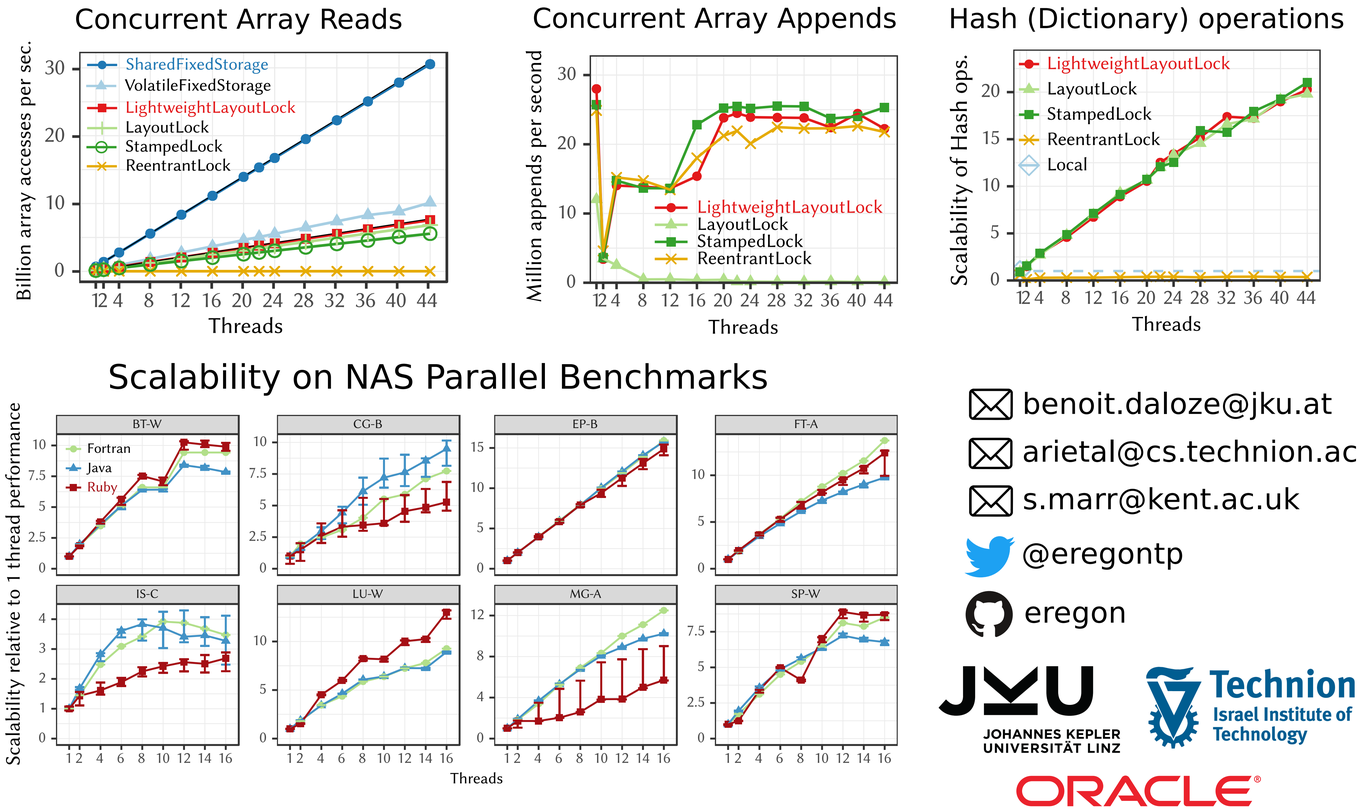

```ruby
array = [1, 2, 3, 4] # int[]

$global = array # => SharedFixedStorage(int[])
$global[0] = $global[-1] # OK

# these migrate to SharedDynamicStorage:
array[1] = Object.new
array << 5
array.delete_at(2)
```

TRUFFLE RUBY

https://github.com/graalvm/truffleruby

Performance and Scalability

Higher is better



benoit.daloze@jku.at
arietal@cs.technion.ac.il
s.marr@kent.ac.uk
@eregontp
eregon

JOHANNES KEPLER UNIVERSITÄT LINZ

Technion Israel Institute of Technology

ORACLE