Thread-Safe Built-in Collections for Dynamic Languages
Benoit Daloze, Arie Tal, Stefan Marr, Hanspeter Mössenböck, Erez Petrank

Thread-Safety Problems

```ruby
class Array
  def initialize
    @data = []
  end

  def append(x)
    @data << x
  end

  def size
    @data.length
  end
end
```

```ruby
MRL, with a GIL: Ruby, with concurrent threads:
```

```ruby
$ruby app.rb
@ concurrent
```

```ruby
ConcurrentError: Detect invalid array contents due to unsynchronized modifications
```

```ruby
Array.new(42, 0.0)
```

Storage Strategies

- **Int**
- **Long**
- **Object**
- **Double**

Tracking sharing of collections

```
Thread 1
```

```
Queue
```

```
Thread 2
```

```
Message
```

```
Node
```

```
"bar"
```

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

Idea: Concurrent Strategies

```
store int
store long
store double
```

```
store Object
```

```
storage transition on sharing
```

```
TRUFFLE RUBY
```

```
https://github.com/graalvm/trufferuby
```

Performance and Scalability ➧ Higher is better

```
Concurrent Array Reads
```

```
Concurrent Array Appends
```

```
Hash (Dictionary) operations
```

```
Scalability on NAS Parallel Benchmarks
```

```
benoit.daloze@ku.at
arietal@cs.technion.ac.il
s.marr@kent.ac.uk
@eregon tp
eregon
```

```
Oracle
```

```
JAYA
```

```
Technion
```

```
University Linz
```