

Ruby - Bug #19917

Segmentation fault or lost objects when using Ractor.select with moved exceptions

10/10/2023 12:17 PM - toy (Ivan Kuchin)

|                 |                                                            |                                                    |
|-----------------|------------------------------------------------------------|----------------------------------------------------|
| Status:         | Closed                                                     |                                                    |
| Priority:       | Normal                                                     |                                                    |
| Assignee:       |                                                            |                                                    |
| Target version: |                                                            |                                                    |
| ruby -v:        | ruby 3.2.2 (2023-03-30 revision e51014f9c0) [x86_64-linux] | Backport: 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN |

**Description**

I stumbled upon loss of messages with exceptions or even Segmentation fault during transfer of exception from Ractor when moving the exception. In versions 3.0 and 3.1 I saw only loss of messages, in 3.2 also Segmentation fault. Loss of messages happens only when using Ractor.select (not Ractor#take), but segmentation fault happens also using Ractor#take.

```
100.times do
  ractor_count = 10

  ractors = Array.new(ractor_count) do |i|
    Ractor.new do
      begin
        raise 'foo'
      rescue => e
        # It is not possible to move exception without duplicating, but also
        # without it showing backtrace errors in 3.0 and is empty in 3.1+
        e = Marshal.load(Marshal.dump(e))
        Ractor.yield e, move: true
      end
      'message got lost'
    end
  end

  ractor_count.times do
    ractor, result = Ractor.select(*ractors)
    p result
    ractors.delete(ractor)
  end
end
```

In 3.0 and 3.1 will contain mix of #<RuntimeError: foo> and "message got lost", in 3.2 may contain both and cause Segmentation fault. Excerpt of output (full output attached) from container running ruby:3.2 (ruby 3.2.2 (2023-03-30 revision e51014f9c0) [x86\_64-linux]):

```
...
#<RuntimeError: RuntimeError>
"message got lost"
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
```

```
#<RuntimeError: RuntimeError>
#<RuntimeError: RuntimeError>
<internal:marshal>:34: [BUG] Segmentation fault at 0x0000000000000000
ruby 3.2.2 (2023-03-30 revision e51014f9c0) [x86_64-linux]

-- Control frame information -----
c:0004 p:0006 s:0019 e:000018 METHOD <internal:marshal>:34
c:0003 p:0022 s:0011 e:000010 RESCUE (irb):11
c:0002 p:0007 s:0007 e:000006 BLOCK (irb):6 [FINISH]
c:0001 p:---- s:0003 e:000002 DUMMY [FINISH]

-- Ruby level backtrace information -----
(irb):6:in `block (3 levels) in <top (required)>\'
(irb):11:in `rescue in block (3 levels) in <top (required)>\'
<internal:marshal>:34:in `load\'

-- Machine register context -----
RIP: 0x00007fabcd1d64490 RBP: 0x0000000000000000 RSP: 0x00007fabbc71b0c0
RAX: 0x0000000000000003 RBX: 0x00007fabbcebee88 RCX: 0x000000000000015f
RDX: 0x0000556127db90f0 RDI: 0x000000000000015f RSI: 0x00007fabcd2068628
R8: 0x0000000000000001 R9: 0x0000000000000020 R10: 0x00005561285633d0
R11: 0x0000000000000001 R12: 0x0000556127dbb710 R13: 0x00007fabcd0601be0
R14: 0x0000000000000018 R15: 0x0000000000000000 EFL: 0x0000000000010206
...
```

## Associated revisions

### Revision 054f56fd3e5bf84e5443896fd1f4e439c2773c60 - 12/19/2023 10:04 PM - ko1 (Koichi Sasada)

moved object should not have a shape ID

fix [Bug #19917]

### Revision 054f56fd3e5bf84e5443896fd1f4e439c2773c60 - 12/19/2023 10:04 PM - ko1 (Koichi Sasada)

moved object should not have a shape ID

fix [Bug #19917]

### Revision 054f56fd - 12/19/2023 10:04 PM - ko1 (Koichi Sasada)

moved object should not have a shape ID

fix [Bug #19917]

## History

### #1 - 10/15/2023 01:02 AM - luke-gru (Luke Gruber)

I'm not in front of a machine that runs ruby right now, but I don't think this has to do with ractors. I was trying things out in the ruby playground and managed to crash ruby with this as well:

```
100.times do |j|
  ractor_count = 10

  ractors = Array.new(ractor_count) do |i|
    Proc.new do
      begin
        raise 'foo'
      rescue => e
        err = Marshal.load(Marshal.dump(e))
      end
      #e
    end
  end

  ractors.each(&:call)
end
```

It looks like it's something related to Marshal.

### #2 - 10/15/2023 04:46 PM - toy (Ivan Kuchin)

I wasn't able to reproduce segmentation fault, but even if segmentation fault is caused by Marshal, it doesn't explain lost messages (when Ractor.select returns a ractor and "message got lost").

**#3 - 12/19/2023 10:29 PM - ko1 (Koichi Sasada)**  
- Status changed from Open to Closed

Applied in changeset [git|054f56fd3e5bf84e5443896fd1f4e439c2773c60](#).

---

moved object should not have a shape ID

fix [Bug [#19917](#)]

**Files**

|                        |         |            |                   |
|------------------------|---------|------------|-------------------|
| segmentation-fault.txt | 27.8 KB | 10/10/2023 | toy (Ivan Kuchin) |
|------------------------|---------|------------|-------------------|