

Ruby - Bug #20112

Ractors not working properly in ruby 3.3.0

01/03/2024 03:51 PM - ariasdiniz (Aria Diniz)

Status:	Closed	Backport: 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN
Priority:	Normal	
Assignee:	ractor	
Target version:		
ruby -v:	3.3.0	

**Description**

I recently installed Ruby 3.3.0, and noticed that some of my scripts that use Ractors started to struggle with performance. After doing some benchmarks, I noticed that, while Ractors seem to be working well on Ruby 3.2.2, they're not working properly on 3.3.0.

I'm using Ubuntu 22.04.3 LTS

Here is the benchmark code:

```
# frozen_string_literal: true

require 'benchmark'

Ractor.new { :warmup } if defined?(Ractor)

Benchmark.bmbm do |x|
  x.report("Thread: ") do
    threads = []
    8.times do |i|
      threads << Thread.new do
        20000000.times do |j|
          ((i * 20000000) + j)**2
        end
      end
    end
    threads.each(&:join)
  end
  x.report("Ractor: ") do
    ractors = []
    0..8.times do |i|
      ractors << Ractor.new(i) do |k|
        20000000.times do |j|
          ((k * 20000000) + j)**2
        end
      end
    end
    ractors.map(&:take)
  end
end
```

Here is the results for Ruby 3.2.2:

```
Rehearsal -----
Thread:   7.666909   0.001091   7.668000 ( 7.675266)
Ractor:   19.318528   0.012017  19.330545 ( 2.505888)
----- total: 26.998545sec
```

	user	system	total	real
Thread:	7.918141	0.004011	7.922152	( 7.928772)
Ractor:	19.366414	0.003954	19.370368	( 2.517993)

Here is the results for Ruby 3.3.0:

Rehearsal -----				
Thread:	8.634152	0.010895	8.645047 (	8.645104)
Ractor:	100.172179	0.035985	100.208164 (	15.213245)
----- total: 108.853211sec				
	user	system	total	real
Thread:	9.451236	0.004002	9.455238 (	9.460132)
Ractor:	118.463294	0.119942	118.583236 (	18.462157)

## History

### #1 - 01/03/2024 10:57 PM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned
- Assignee set to ko1 (Koichi Sasada)

### #2 - 05/08/2025 10:38 PM - jhawthorn (John Hawthorn)

- Assignee changed from ko1 (Koichi Sasada) to ractor

### #3 - 05/19/2025 08:34 AM - byroot (Jean Boussier)

- Description updated

### #4 - 05/19/2025 08:57 AM - etienne (Étienne Barrié)

I can't repro on my mac so it seems Linux only.

It has been corrected on master since both 3.5.0-preview1 and master are now faster with ractors than with threads.

```
ruby 3.5.0dev (2025-05-19T04:27:50Z master 40c957ba21) +PRISM [aarch64-linux]
-:21: warning: possibly useless use of .. in void context
Rehearsal -----
-:5: warning: Ractor is experimental, and the behavior may change in future versions of Ruby! Also there are m
any implementation issues.
Thread: 5.074042 0.001007 5.075049 ( 5.076653)
Ractor: 5.375332 0.000005 5.375337 ( 1.548004)
----- total: 10.450386sec

      user      system      total      real
Thread: 5.098492 0.001000 5.099492 ( 5.098067)
Ractor: 5.254631 0.001000 5.255631 ( 1.545433)
```

### #5 - 05/19/2025 08:58 AM - byroot (Jean Boussier)

- Status changed from Assigned to Closed