Ruby - Bug #17489

Ractor segfaults

12/28/2020 09:23 PM - kirs (Kir Shatrov)

Status:	Closed		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:			
ruby -v:	3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin19]	**Backport:**	2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN

Description

I was able to boil it down to this script:

```ruby
require 'bundler/inline'

gemfile do
  source 'https://rubygems.org'
  gem "sinatra", "~> 2.1"
end

require 'sinatra'

class App < Sinatra::Base
  get '/' do
    200
  end
end

module Sinatra
  class Base
    class << self
      # patch it to avoid hitting other unsafe things
      def setup_default_middleware(builder)
        builder.use Rack::NullLogger
      end
    end
  end
end

workers = 2.times.map do
  Ractor.new { App.new }
end

Ractor.select(*workers)
```

Even though the code doesn't make a lot of sense and Sinatra has other stuff that's not compatible with Ractor, I would not expect user-level Ruby code to segfault.

I have attached the crash report and stderr output.

Related issues:

Related to Ruby - Bug #17529: Ractor Segfaults with GC enabled	**Closed**
Is duplicate of Ruby - Bug #18120: Deadlock and segfault when using autoload ...	**Closed**

History

#1 - 12/28/2020 09:23 PM - kirs (Kir Shatrov)

- ruby -v set to 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin19]

#2 - 12/28/2020 09:28 PM - kirs (Kir Shatrov)

- Description updated

#3 - 01/05/2021 01:34 AM - ko1 (Koichi Sasada)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

Thank you. I can confirm the reproducing. This issue is maybe from the lack of synchronization for the constant table.

#4 - 01/05/2021 09:25 PM - inversion (Yura Babak)

During my Ractors tests I had segfaults many times and easy to reproduce (in my environment).
The same code crashes on Linux and Windows10 (sometimes with other numbers of iterations or workers).
More Ractors seems to increase the segfault rate.
And if I add GC.disable — no crash.

So I was looking for a place where to add a report with my code samples.
This issue looks like a good place for this.

Test cases:
crash_reused_sha2.rb
https://gist.github.com/Inversion-des/fcc7dbb2d9a77a7dba540c29077e5ebf
(90% crash, if I uncomment GC.disable — no crash)

crash_encrypt_big_data_often.rb
https://gist.github.com/Inversion-des/a9dbde4eacfa859e5bd7cc8fd8a8049f
(50% crash, again uncomment GC.disable — no crash)

Win10_crash_report.txt
https://gist.github.com/Inversion-des/c79ea00407c579b83ebbe00541e20c45

If it doesn't crash for you — try to increase number of iterations and data size.

My environment:
Linux CentOS 7.9 (VPS)
4-core CPU 2.6GHz
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-linux]
Windows 10
2-core CPU 3.6GHz
ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x64-mingw32]

#5 - 01/12/2021 01:42 AM - marcandre (Marc-Andre Lafortune)

- Related to Bug #17529: Ractor Segfaults with GC enabled added

#6 - 03/25/2022 11:29 AM - wanabe (_ wanabe)

It seems be a autoload on non-main Ractor problem [Bug #18120].
This operation is prohibited after
https://bugs.ruby-lang.org/projects/ruby-master/repository/git/revisions/397a509b6d0d1470df8c290d7c4adef78f1532ee and does not result in a BUG
thereafter.

#7 - 03/26/2022 09:29 AM - wanabe (_ wanabe)

- Is duplicate of Bug #18120: Deadlock and segfault when using autoload in Ractor added

#8 - 03/26/2022 09:50 AM - kirs (Kir Shatrov)

wanabe (_ wanabe) wrote in #note-6:

> It seems be a autoload on non-main Ractor problem [Bug #18120].
> This operation is prohibited after
> https://bugs.ruby-lang.org/projects/ruby-master/repository/git/revisions/397a509b6d0d1470df8c290d7c4adef78f1532ee and does not result in a
> BUG thereafter.

That's great news!

I wish I knew how to close this ticket. Not sure I have permissions to do that...?

#9 - 03/26/2022 08:36 PM - wanabe (_ wanabe)

- Status changed from Assigned to Closed

kirs (Kir Shatrov) wrote in #note-8:

I wish I knew how to close this ticket. Not sure I have permissions to do that...?

OK, I'll close it.

Files

segfault.dump	44.8 KB	12/28/2020	kirs (Kir Shatrov)
ruby_2020-12-29-002108_Kirs-MacBook-Pro-2.crash	27.9 KB	12/28/2020	kirs (Kir Shatrov)