

Ruby - Bug #17826

Ractor#take hangs if used in multiple Threads

04/25/2021 11:07 AM - kukunin (Sergiy Kukunin)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
ruby -v:	ruby 3.0.1p64 (2021-04-05 revision 0fb782ee38) [x86_64-linux]	Backport: 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN
Description Hello there. I was playing with Ractors (the awesome technology and the big leap for Ruby) and encountered weird behavior. I tried to schedule and run Ractors in multiple threads, and found out, that Ractor#take hangs even if the ractor is finished. Here is code to reproduce: <pre>Array.new(2) do n Thread.new do r = Ractor.new do sleep 0.001 end r.take puts "thread #{n} finished" end end.each(&:join)</pre> puts 'done' The output is just "thread 0 finished" and the process hangs forever. Sometimes the second thread exists first, and the first freezes. Thank you for your time. Hopefully, it's a valid bug report, not me just misusing the feature =)		

History

#1 - 04/25/2021 12:53 PM - kukunin (Sergiy Kukunin)

It reproduces on the master branch as well (git commit cc5bab80e4b1124e30a8dc220bdc80aaa539486).

If I connect with a debugger to the process, there are two threads left (out of 5 at the most, 1 main, 2 threads, and 2 ractors)

```
(gdb) info threads
  Id   Target Id               Frame
*  1   LWP 3345410 "miniruby" 0x00007f0f35d5e47e in ppoll () from /usr/lib/libc.so.6
   2   LWP 3345412 "bug.rb:2" 0x00007f0f3607c9ba in __futex_abstimed_wait_common64 () from /usr/lib/libpthread.so.0

(gdb) bt
#0  0x00007f0f35d5e47e in ppoll () from /usr/lib/libc.so.6
#1  0x0000555cb72ca246 in rb_sigwait_sleep (th=th@entry=0x555cb940bf70, sigwait_fd=sigwait_fd@entry=3, rel=rel@entry=0x0) at thread_pthread.c:2073
#2  0x0000555cb72ca4be in native_sleep (th=th@entry=0x555cb940bf70, rel=rel@entry=0x0) at thread_pthread.c:2194
#3  0x0000555cb72cc079 in thread_join_sleep (arg=arg@entry=140720351805792) at thread.c:1191
#4  0x0000555cb7175e38 in rb_ensure (b_proc=b_proc@entry=0x555cb72cbecc <thread_join_sleep>, data1=data1@entry=140720351805792, e_proc=e_proc@entry=0x555cb72c3f18 <remove_from_join_list>, data2=data2@entry=140720351805792) at eval.c:1166
#5  0x0000555cb72c4f79 in thread_join (target_th=0x555cb94cb140, timeout=timeout@entry=8) at thread.c:1241
#6  0x0000555cb72c6dd3 in thread_join_m (argc=<error reading variable: value has been optimized out>, argv=<error reading variable: value has been optimized out>, self=139703308083720, self@entry=<error reading variable: value has been optimized out>) at thread.c:1334
#7  0x0000555cb72f0570 in ractor_safe_call_cfunc_m1 (recv=<optimized out>, argc=<optimized out>, argv=<optimized out>, func=<optimized out>) at vm_insnhelper.c:2753
```

```

#8 0x0000555cb72fb504 in vm_call0_cfunc_with_frame (ec=ec@entry=0x555cb940c7d0, calling=calling@entry=0x7ffc02950310, argv=argv@entry=0x7ffc02950430) at vm_eval.c:135
#9 0x0000555cb72fb6e1 in vm_call0_cfunc (ec=ec@entry=0x555cb940c7d0, calling=calling@entry=0x7ffc02950310, argv=argv@entry=0x7ffc02950430) at vm_eval.c:149
#10 0x0000555cb730e87f in vm_call0_body (ec=ec@entry=0x555cb940c7d0, calling=calling@entry=0x7ffc02950310, argv=argv@entry=0x7ffc02950430) at vm_eval.c:182
#11 0x0000555cb7310b14 in vm_call0_cc (kw_splat=0, cc=0x7f0f360d60f0, argv=0x7ffc02950430, argc=0, id=<optimized out>, recv=139703308083720, ec=0x555cb940c7d0)
    at vm_eval.c:72
#12 rb_call0 (ec=0x555cb940c7d0, recv=139703308083720, mid=<optimized out>, argc=0, argv=0x7ffc02950430, argv@entry=0x8, call_scope=<optimized out>,
    self=139703308084200) at vm_eval.c:520
#13 0x0000555cb73114d3 in rb_call (scope=<optimized out>, argv=0x8, argc=<optimized out>, mid=<optimized out>, recv=<optimized out>) at vm_eval.c:844
#14 rb_funcall_with_block_kw (recv=<optimized out>, mid=<optimized out>, argc=argc@entry=0, argv=argv@entry=0x7ffc02950430, passed_procval=passed_procval@entry=8,
    kw_splat=kw_splat@entry=0) at vm_eval.c:1191
#15 0x0000555cb72bc3b0 in rb_sym_proc_call (mid=<optimized out>, argc=argc@entry=1, argv=argv@entry=0x7ffc02950428, kw_splat=kw_splat@entry=0,
    passed_proc=passed_proc@entry=8) at string.c:11154
#16 0x0000555cb7302f07 in vm_yield_with_symbol (ec=ec@entry=0x555cb940c7d0, symbol=3354892, argc=argc@entry=1, argv=argv@entry=0x7ffc02950428,
    kw_splat=kw_splat@entry=0, block_handler=block_handler@entry=0) at vm_insnhelper.c:3758
#17 0x0000555cb731686e in invoke_block_from_c_bh (force_blockarg=<optimized out>, is_lambda=0, cref=<optimized out>, passed_block_handler=<optimized out>,
    kw_splat=<optimized out>, argv=<optimized out>, argc=<optimized out>, block_handler=<optimized out>, ec=<optimized out>) at vm.c:1359
#18 vm_yield (kw_splat=0, argv=0x7ffc02950428, argc=1, ec=0x555cb940c7d0) at vm.c:1395
#19 rb_yield_0 (argv=0x7ffc02950428, argc=1) at vm_eval.c:1343
#20 rb_yield (val=<optimized out>) at vm_eval.c:1359
#21 0x0000555cb70f8f4f in rb_ary_each (ary=139703308084200, ary@entry=<error reading variable: value has been optimized out>) at array.c:2529
#22 0x0000555cb72f057b in ractor_safe_call_cfunc_0 (recv=<optimized out>, argc=<optimized out>, argv=<optimized out>, func=<optimized out>) at vm_insnhelper.c:2760
#23 0x0000555cb72fb1ab in vm_call_cfunc_with_frame (ec=ec@entry=0x555cb940c7d0, reg_cfp=reg_cfp@entry=0x7f0f356bcfa0, calling=calling@entry=0x7ffc02950620)
    at vm_insnhelper.c:2943
#24 0x0000555cb72fc58c in vm_call_cfunc (ec=ec@entry=0x555cb940c7d0, reg_cfp=reg_cfp@entry=0x7f0f356bcfa0, calling=calling@entry=0x7ffc02950620)
    at vm_insnhelper.c:2964
#25 0x0000555cb730bf35 in vm_call_method_each_type (ec=0x555cb940c7d0, cfp=0x7f0f356bcfa0, calling=calling@entry=0x7ffc02950620) at vm_insnhelper.c:3433
#26 0x0000555cb730ce01 in vm_call_method (calling=0x7ffc02950620, cfp=0x7f0f356bcfa0, ec=0x555cb940c7d0) at vm_insnhelper.c:3526
#27 vm_call_general (ec=0x555cb940c7d0, reg_cfp=0x7f0f356bcfa0, calling=0x7ffc02950620) at vm_insnhelper.c:3569
#28 0x0000555cb7304bc6 in vm_sendish (ec=0x555cb940c7d0, reg_cfp=0x7f0f356bcfa0, cd=0x555cb94f0990, block_handler=<optimized out>, method_explorer=<optimized out>)
#29 0x0000555cb7309726 in vm_exec_core (ec=ec@entry=0x555cb940c7d0, initial=<optimized out>) at insns.def:754
#30 0x0000555cb730b6c7 in rb_vm_exec (ec=ec@entry=0x555cb940c7d0, mjit_enable_p=mjit_enable_p@entry=true) at vm.c:2160
#31 0x0000555cb730d593 in rb_iseq_eval_main (iseq=iseq@entry=0x7f0f360d6988) at vm.c:2417
#32 0x0000555cb7173da5 in rb_ec_exec_node (ec=ec@entry=0x555cb940c7d0, n=n@entry=0x7f0f360d6988) at eval.c:317
#33 0x0000555cb7177d59 in ruby_run_node (n=0x7f0f360d6988) at eval.c:375
#34 0x0000555cb70f83bb in main (argc=<optimized out>, argv=<optimized out>) at ./main.c:47

```

(gdb) thread apply 2 bt

Thread 2 (LWP 3345412 "bug.rb:2"):

```

#0 0x00007f0f3607c9ba in __futex_abstimed_wait_common64 () from /usr/lib/libpthread.so.0
#1 0x00007f0f36076260 in pthread_cond_wait@@GLIBC_2.3.2 () from /usr/lib/libpthread.so.0
#2 0x0000555cb72c88c5 in rb_native_cond_wait (cond=cond@entry=0x555cb940ca30, mutex=mutex@entry=0x555cb940ca08) at thread_pthread.c:542
#3 0x0000555cb724b942 in ractor_cond_wait (r=r@entry=0x555cb940c9e0) at ractor.c:112
#4 0x0000555cb724b992 in ractor_sleep_wo_gvl (ptr=ptr@entry=0x555cb940c9e0) at ractor.c:527
#5 0x0000555cb72ccb86 in rb_nogvl (func=func@entry=0x555cb724b947 <ractor_sleep_wo_gvl>, data1=data1@entry=0x555cb940c9e0, ubf=<optimized out>, ubf@entry=0x555cb724b31c <ractor_sleep_interrupt>, data2=data2@entry=0x555cb940c9e0, flags=flags@entry=3) at thread.c:1673
#6 0x0000555cb724b84e in ractor_sleep (ec=ec@entry=0x555cb94cb330, cr=cr@entry=0x555cb940c9e0) at ractor.c:590
#7 0x0000555cb724eab7 in ractor_select (ec=0x555cb94cb330, ec@entry=<error reading variable: value has been optimized out>, rs=<optimized out>, rs_len=rs_len@entry=1, yielded_value=yielded_value@entry=52, move=move@entry=false, ret_r=ret_r@entry=0x7f0f31de59f0) at ractor.c:1235
#8 0x0000555cb724ec12 in ractor_take (ec=<error reading variable: value has been optimized out>, r=<optimized out>) at ractor.c:1335
#9 0x0000555cb724ec39 in builtin_inline_class_694 (ec=<error reading variable: value has been optimized out>,

```

```
self=<error reading variable: value has been optimized out>) at ractor.c:290
#10 0x0000555cb72f088b in builtin_invoker0 (ec=<optimized out>, self=<optimized out>, argv=<optimized out>, fu
ncptr=<optimized out>) at vm_insnhelper.c:5429
#11 0x0000555cb72f325e in invoke_bf (argv=0x0, bf=0x555cb7438960 <ractor_table+288>, reg_cfp=0x7f0f31ee5cb8, e
c=0x555cb94cb330) at vm_insnhelper.c:5569
#12 vm_invoke_builtin_delegate (ec=ec@entry=0x555cb94cb330, cfp=cfp@entry=0x7f0f31ee5cb8, bf=bf@entry=0x555cb7
438960 <ractor_table+288>, start_index=start_index@entry=0) at vm_insnhelper.c:5593
#13 0x0000555cb730b1ba in vm_exec_core (ec=ec@entry=0x555cb94cb330, initial=<optimized out>) at insns.def:1477
#14 0x0000555cb730b6c7 in rb_vm_exec (ec=ec@entry=0x555cb94cb330, mjit_enable_p=mjit_enable_p@entry=true) at v
m.c:2160
#15 0x0000555cb730e4c8 in invoke_block (opt_pc=<optimized out>, type=572653569, cref=0x0, captured=0x555cb94cb
570, self=139703296263320, iseq=0x7f0f360d6820, ec=0x555cb94cb330) at vm.c:1260
#16 invoke_iseq_block_from_c (me=0x0, is_lambda=<optimized out>, cref=0x0, passed_block_handler=0, kw_splat=0,
argv=0x7f0f31de5c80, argc=0, self=139703296263320, captured=0x555cb94cb570, ec=0x555cb94cb330) at vm.c:1332
#17 invoke_block_from_c_proc (me=0x0, is_lambda=<optimized out>, passed_block_handler=0, kw_splat=0, argv=0x7f
0f31de5c80, argc=0, self=139703296263320, proc=0x7f0f360d6820, ec=0x555cb94cb330) at vm.c:1432
#18 vm_invoke_proc (ec=ec@entry=0x555cb94cb330, proc=proc@entry=0x555cb94cb570, self=139703296263320, argc=arg
c@entry=0, argv=argv@entry=0x7f0f31de5c80, kw_splat=kw_splat@entry=0, passed_block_handler=0) at vm.c:1461
#19 0x0000555cb730e79a in rb_vm_invoke_proc (ec=0x555cb94cb330, proc=proc@entry=0x555cb94cb570, argc=argc@entr
y=0, argv=argv@entry=0x7f0f31de5c80, kw_splat=0, passed_block_handler=passed_block_handler@entry=0) at vm.c:14
82
#20 0x0000555cb72cd091 in thread_do_start_proc (th=th@entry=0x555cb94cb140) at thread.c:738
#21 0x0000555cb72cd202 in thread_do_start (th=0x555cb94cb140) at thread.c:759
#22 0x0000555cb72cd468 in thread_start_func_2 (th=th@entry=0x555cb94cb140, stack_start=<optimized out>) at thr
ead.c:824
#23 0x0000555cb72cd882 in thread_start_func_1 (th_ptr=<optimized out>) at thread_pthread.c:1033
#24 0x00007f0f36070299 in start_thread () from /usr/lib/libpthread.so.0
#25 0x00007f0f35d69053 in clone () from /usr/lib/libc.so.6
```

Hope this helps

#2 - 12/02/2021 07:21 PM - ko1 (Koichi Sasada)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

#3 - 12/14/2021 04:29 PM - ko1 (Koichi Sasada)

Yes, now it doesn't work correctly. Sorry I'll try to fix this issue on Ruby 3.2.

#4 - 12/15/2021 01:30 PM - kukunin (Sergiy Kukunin)

Thank you for the review, and I'm glad to provide a confirmed bug report.

Volodya Sveredyuk and I invite you to give me a talk in our great and cozy local Ruby community in Lviv, Ukraine - Pivorak. Aaron Patterson and José Valim already presented at Pivorak, and we would be super glad to accommodate more superstars that make the community better.

#5 - 05/24/2023 09:35 PM - kukunin (Sergiy Kukunin)

I have tried the latest stable Ruby version 3.2.2 and the bug is still here

ruby 3.2.2 (2023-03-30 revision e51014f9c0) [arm64-darwin22]

#6 - 08/24/2023 09:32 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Assigned to Closed

This appears fixed in Ruby 3.3.0-preview1, though it doesn't work in Ruby 3.2.2.