Ruby - Bug #21034

try to mark T_NONE object error after upgrading to 3.4.1

01/14/2025 12:35 AM - travisbell (Travis Bell)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 3.4.1 (2024-12-25 revision 48d4efcb85) +PRISM [x86_64-linux]	**Backport:**	3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

Hi everyone,

I noticed we started having some workers crash after upgrading to 3.4.1. I tried grabbing a core file and got this output. Is it enough to figure out what's going on? If not, I can reproduce this fairly easily, so I can grab more debug info if needed.

```
#0  __pthread_kill_implementation (no_tid=0, signo=6, threadid=<optimized out>) at ./nptl/pthread_
kill.c:44
        tid = <optimized out>
        ret = 0
        pd = <optimized out>
        old_mask = {__val = {2337500343188860976}}
        ret = <optimized out>
        pd = <optimized out>
        old_mask = {__val = {<optimized out>}}
        ret = <optimized out>
        tid = <optimized out>
        ret = <optimized out>
        resultvar = <optimized out>
        resultvar = <optimized out>
        __arg3 = <optimized out>
        __arg2 = <optimized out>
        __arg1 = <optimized out>
        _a3 = <optimized out>
        _a2 = <optimized out>
        _a1 = <optimized out>
        __futex = <optimized out>
        resultvar = <optimized out>
        __arg3 = <optimized out>
        __arg2 = <optimized out>
        __arg1 = <optimized out>
        _a3 = <optimized out>
        _a2 = <optimized out>
        _a1 = <optimized out>
        __futex = <optimized out>
        __private = <optimized out>
        __oldval = <optimized out>
#1  __pthread_kill_internal (signo=6, threadid=<optimized out>) at ./nptl/pthread_kill.c:78
#2  __GI___pthread_kill (threadid=<optimized out>, signo=signo@entry=6) at ./nptl/pthread_kill.c:8
9
#3  0x00007facb872d26e in __GI_raise (sig=sig@entry=6) at ../sysdeps/posix/raise.c:26
        ret = <optimized out>
#4  0x00007facb87108ff in __GI_abort () at ./stdlib/abort.c:79
        save_stage = 1
        act = {__sigaction_handler = {sa_handler = 0x20, sa_sigaction = 0x20}, sa_mask = {__val =
{2314885530818453536, 2314885530818453536, 6733551554292031520, 3558800716248528650, 7365405400577
881908, 3474865974218745190, 2337500343188860976, 3472328296227680304, 3467824696768081952, 231488
5530818453536, 2314885530818453536, 2314885530818453536, 6732726843261788192, 62, 1802092839518680
5760, 140378616593424}}, sa_flags = -1198603040, sa_restorer = 0x7facb8f11b0c}
#5  0x00007facb8a9686a in die () at error.c:1071
#6  rb_bug (fmt=fmt@entry=0x7facb8f11b0c "try to mark T_NONE object") at error.c:1095
        args = {{gp_offset = 8, fp_offset = 48, overflow_arg_area = 0x7fac71917450, reg_save_area
```

```
= 0x7fac71917380}}
#7  0x00007facb8b98bee in gc_mark (objspace=0x7facb8072000, obj=140378889836960) at gc/default/def
ault.c:4463
        __func__ = {<optimized out>, <optimized out>, <optimized out>, <optimized out>, <optimized
 out>, <optimized out>, <optimized out>, <optimized out>}
#8  gc_mark (obj=140378889836960, objspace=0x7facb8072000) at gc/default/default.c:4443
        __func__ = {<optimized out>, <optimized out>, <optimized out>, <optimized out>, <optimized
 out>, <optimized out>, <optimized out>, <optimized out>}
#9  gc_mark_and_pin (obj=140378889836960, objspace=0x7facb8072000) at gc/default/default.c:4489
#10 rb_gc_impl_mark_and_pin (obj=140378889836960, objspace_ptr=0x7facb8072000) at gc/default/defau
lt.c:4521
        objspace = 0x7facb8072000
        objspace = <optimized out>
#11 gc_mark_and_pin_internal (obj=140378889836960) at gc.c:2204
        vm = <optimized out>
        objspace = 0x7facb8072000
        __func__ = {<optimized out> <repeats 25 times>}
        vm = <optimized out>
        objspace = <optimized out>
        mark_func_data = <optimized out>
#12 gc_mark_and_pin_internal (obj=140378889836960) at gc.c:2202
        __func__ = {<optimized out> <repeats 25 times>}
        vm = <optimized out>
        objspace = <optimized out>
        mark_func_data = <optimized out>
#13 rb_gc_mark_vm_stack_values (n=62, values=0x7fac6ca72000) at gc.c:2291
        i = 29
#14 0x00007facb8d6c339 in rb_execution_context_mark (ec=0x7fac803405d0) at vm.c:3404
        p = <optimized out>
        sp = <optimized out>
        cfp = 0x7fac6ca91d98
        limit_cfp = 0x7fac6ca92000
#15 0x00007facb8b4810c in cont_mark (ptr=0x7fac80340580) at cont.c:1011
        cont = 0x7fac80340580
        cont = <optimized out>
#16 fiber_mark (ptr=0x7fac80340580) at cont.c:1140
        fiber = 0x7fac80340580
#17 0x00007facb8b9d56d in gc_mark_children (obj=140378344077120, objspace=0x7facb8072000) at gc/de
fault/default.c:4647
#18 gc_mark_stacked_objects (incremental=1, count=4478, objspace=0x7facb8072000) at gc/default/def
ault.c:4668
--Type <RET> for more, q to quit, c to continue without paging--
        mstack = 0x7facb8072310
        obj = 140378344077120
        marked_slots_at_the_beginning = 484361
        popped_count = <optimized out>
        mstack = <optimized out>
        obj = <optimized out>
        marked_slots_at_the_beginning = <optimized out>
        popped_count = <optimized out>
#19 gc_mark_stacked_objects_incremental (count=4478, objspace=0x7facb8072000) at gc/default/defaul
t.c:4700
#20 gc_marks_step (slots=4478, objspace=0x7facb8072000) at gc/default/default.c:5735
        marking_finished = false
        marking_finished = <optimized out>
        __func__ = {<optimized out> <repeats 14 times>}
#21 gc_marks_continue (heap=0x7facb8072028, objspace=0x7facb8072000) at gc/default/default.c:5755
        marking_finished = true
        __func__ = {<optimized out> <repeats 18 times>}
        marking_finished = <optimized out>
#22 gc_continue (objspace=0x7facb8072000, heap=0x7facb8072028) at gc/default/default.c:2073
        lock_lev = 0
#23 0x00007facb8ba0ce2 in newobj_cache_miss (objspace=objspace@entry=0x7facb8072000, cache=cache@e
ntry=0x7facb807c1e0, heap_idx=heap_idx@entry=0, vm_locked=<optimized out>, vm_locked@entry=false)
at gc/default/default.c:2414
        heap = 0x7facb8072028
        obj = 0
```

```
        lev = 1
        unlock_vm = true
#24 0x00007facb8ba20e2 in newobj_alloc (vm_locked=false, heap_idx=0, cache=0x7facb807c1e0, objspac
e=0x7facb8072000) at gc/default/default.c:2447
        obj = <optimized out>
        heap = <optimized out>
        obj = <optimized out>
        heap = <optimized out>
        __func__ = {<optimized out> <repeats 13 times>}
#25 rb_gc_impl_new_obj (alloc_size=16, wb_protected=true, v3=0, v2=0, v1=0, flags=8199, klass=1403
79367216160, cache_ptr=0x7facb807c1e0, objspace_ptr=0x7facb8072000) at gc/default/default.c:2533
        obj = <optimized out>
        objspace = 0x7facb8072000
        heap_idx = 0
        cache = 0x7facb807c1e0
        obj = <optimized out>
        objspace = <optimized out>
        heap_idx = <optimized out>
        cache = <optimized out>
        cnt = <optimized out>
        i = <optimized out>
#26 newobj_of (size=16, wb_protected=true, v3=0, v2=0, v1=0, flags=8199, klass=140379367216160, cr
=0x7facb801e400) at gc.c:984
        obj = <optimized out>
        obj = <optimized out>
        lev = <optimized out>
#27 rb_wb_protected_newobj_of (ec=ec@entry=0x7fac803c97d0, klass=klass@entry=140379367216160, flag
s=flags@entry=8199, size=size@entry=16) at gc.c:1013
        __func__ = {<optimized out> <repeats 26 times>}
#28 0x00007facb8aaeda0 in ec_ary_alloc_embed (capa=0, klass=140379367216160, ec=<optimized out>) a
t array.c:790
        size = 16
        ary = <optimized out>
        size = <optimized out>
        ary = <optimized out>
#29 ec_ary_new (capa=0, klass=140379367216160, ec=<optimized out>) at array.c:824
        ary = <optimized out>
        ary = <optimized out>
        dtrace_line = <optimized out>
        dtrace_file = <optimized out>
#30 rb_ec_ary_new_from_values (ec=<optimized out>, n=0, elts=0x0) at array.c:843
        ary = <optimized out>
#31 0x00007fac840bd709 in ??? ()
#32 0x00007fac803c97d0 in ??? ()
#33 0x0000000000000c61 in ??? ()
#34 0x00007fac803c9700 in ??? ()
#35 0x00007facb8d76f98 in jit_exec_exception (ec=<optimized out>) at vm.c:509
        func = <optimized out>
        func = <optimized out>
#36 vm_exec_loop (result=<optimized out>, tag=<optimized out>, state=<optimized out>, ec=<optimize
d out>) at vm.c:2612
        vm_loop_start = <optimized out>
        vm_loop_start = <optimized out>
#37 rb_vm_exec (ec=0x7fac803c97d0) at vm.c:2589
        _ec = 0x7fac803c97d0
        _tag = {tag = 36, retval = 4, buf = {0x0, 0x7facb8d77040 <rb_vm_exec+400>, 0x0, 0x7fac7191
76f0, 0x0}, prev = 0x7fac71917bd0, state = RUBY_TAG_NONE, lock_rec = 0}
        state = RUBY_TAG_NONE
        result = <optimized out>
#38 0x00007facb8d7d454 in vm_call0_cc (ec=0x7fac803c97d0, recv=<optimized out>, id=3169, argc=<opt
imized out>, argv=<optimized out>, cc=<optimized out>, kw_splat=0) at /tmp/ruby-build.202412162350
58.18.2gDQPw/ruby-3.4.0-rc1/vm_eval.c:101
--Type <RET> for more, q to quit, c to continue without paging--
        flags = <optimized out>
        use_argv = <optimized out>
        av = {140378344075280, 140379812157720}
        calling = {cd = 0x7fac719177c0, cc = 0x7fac8086be90, block_handler = 0, recv = 14037872472
```

```
5520, argc = 1, kw_splat = false, heap_argv = 0}
#39 0x00007facb8d80cc5 in rb_call0 (ec=0x7fac803c97d0, recv=recv@entry=140378724725520, mid=mid@en
try=3169, argc=argc@entry=1, argv=argv@entry=0x7fac719181f8, call_scope=call_scope@entry=CALL_FCAL
L, self=140379119115560) at /tmp/ruby-build.20241216235058.18.2gDQPw/ruby-3.4.0-rc1/vm_eval.c:554
        call_status = <optimized out>
        scope = <optimized out>
        kw_splat = <optimized out>
        ci = {flags = 110618, kwarg = 0x0, mid = 3169, flag = 4, argc = 1}
        cc = <optimized out>
#40 0x00007facb8d815e6 in rb_call (scope=CALL_FCALL, argv=0x7fac719181f8, argc=1, mid=3169, recv=1
40378724725520) at /tmp/ruby-build.20241216235058.18.2gDQPw/ruby-3.4.0-rc1/vm_eval.c:873
        ec = <optimized out>
#41 rb_funcallv_kw (recv=recv@entry=140378724725520, mid=mid@entry=3169, argc=argc@entry=1, argv=a
rgv@entry=0x7fac719181f8, kw_splat=<optimized out>) at /tmp/ruby-build.20241216235058.18.2gDQPw/ru
by-3.4.0-rc1/vm_eval.c:1070
#42 0x00007facb8b7d863 in rb_obj_call_init_kw (obj=obj@entry=140378724725520, argc=argc@entry=1, a
rgv=argv@entry=0x7fac719181f8, kw_splat=<optimized out>) at eval.c:1731
#43 0x00007facb8c1d4cc in rb_class_new_instance_pass_kw (argc=1, argv=0x7fac719181f8, klass=140379
119115560) at object.c:2175
        obj = 140378724725520
#44 0x00007fac842295cb in ??? ()
#45 0x00007fac71937e78 in ??? ()
#46 0x0000000000000024 in ??? ()
#47 0x00007fac803c97d0 in ??? ()
#48 0x00007facb8d72f2a in vm_exec_core (ec=0x6e2, ec@entry=0x7fac803c97d0) at /tmp/ruby-build.2024
1216235058.18.2gDQPw/ruby-3.4.0-rc1/insns.def:852
        func = 0x0
        bh = 140379806001948
        cd = 0x7fac803c97d0
        blockiseq = 0x6
        leaf = <optimized out>
        val = 140378862557136
        reg_pc = 0x7fac980adde0
        reg_cfp = 0x7fac71937e78
        insns_address_table = {0x7facb8d7019e <vm_exec_core+990>, 0x7facb8d7012a <vm_exec_core+874
>, 0x7facb8d700b9 <vm_exec_core+761>, 0x7facb8d70575 <vm_exec_core+1973>, 0x7facb8d704f3 <vm_exec_
core+1843>, 0x7facb8d70474 <vm_exec_core+1716>, 0x7facb8d703b1 <vm_exec_core+1521>, 0x7facb8d70658
 <vm_exec_core+2200>, 0x7facb8d705f9 <vm_exec_core+2105>, 0x7facb8d702e0 <vm_exec_core+1312>, 0x7f
acb8d701c8 <vm_exec_core+1032>, 0x7facb8d70ba4 <vm_exec_core+3556>, 0x7facb8d70b57 <vm_exec_core+3
479>, 0x7facb8d70af1 <vm_exec_core+3377>, 0x7facb8d70a7c <vm_exec_core+3260>, 0x7facb8d70a37 <vm_e
xec_core+3191>, 0x7facb8d709f7 <vm_exec_core+3127>, 0x7facb8d709ba <vm_exec_core+3066>, 0x7facb8d7
097d <vm_exec_core+3005>, 0x7facb8d70940 <vm_exec_core+2944>, 0x7facb8d708d5 <vm_exec_core+2837>,
0x7facb8d70889 <vm_exec_core+2761>, 0x7facb8d7083a <vm_exec_core+2682>, 0x7facb8d707da <vm_exec_co
re+2586>, 0x7facb8d7078c <vm_exec_core+2508>, 0x7facb8d70705 <vm_exec_core+2373>, 0x7facb8d706c3 <
vm_exec_core+2307>, 0x7facb8d728ab <vm_exec_core+10987>, 0x7facb8d72849 <vm_exec_core+10889>, 0x7f
acb8d72910 <vm_exec_core+11088>, 0x7facb8d720fa <vm_exec_core+9018>, 0x7facb8d71ef1 <vm_exec_core+
8497>, 0x7facb8d71e5b <vm_exec_core+8347>, 0x7facb8d71df5 <vm_exec_core+8245>, 0x7facb8d71d95 <vm_
exec_core+8149>, 0x7facb8d71d33 <vm_exec_core+8051>, 0x7facb8d71cde <vm_exec_core+7966>, 0x7facb8d
71c62 <vm_exec_core+7842>, 0x7facb8d71c10 <vm_exec_core+7760>, 0x7facb8d71be1 <vm_exec_core+7713>,
 0x7facb8d71b9e <vm_exec_core+7646>, 0x7facb8d71b45 <vm_exec_core+7557>, 0x7facb8d71b09 <vm_exec_c
ore+7497>, 0x7facb8d723bf <vm_exec_core+9727>, 0x7facb8d72374 <vm_exec_core+9652>, 0x7facb8d72328
<vm_exec_core+9576>, 0x7facb8d722ee <vm_exec_core+9518>, 0x7facb8d72277 <vm_exec_core+9399>, 0x7fa
cb8d7220d <vm_exec_core+9293>, 0x7facb8d721b6 <vm_exec_core+9206>, 0x7facb8d7213f <vm_exec_core+90
87>, 0x7facb8d72621 <vm_exec_core+10337>, 0x7facb8d72506 <vm_exec_core+10054>, 0x7facb8d724c2 <vm_
exec_core+9986>, 0x7facb8d72473 <vm_exec_core+9907>, 0x7facb8d72747 <vm_exec_core+10631>, 0x7facb8
d72687 <vm_exec_core+10439>, 0x7facb8d6fe30 <vm_exec_core+112>, 0x7facb8d727b9 <vm_exec_core+10745
>, 0x7facb8d70d00 <vm_exec_core+3904>, 0x7facb8d71ab2 <vm_exec_core+7410>, 0x7facb8d71a5b <vm_exec
_core+7323>, 0x7facb8d73741 <vm_exec_core+14721>, 0x7facb8d71a04 <vm_exec_core+7236>, 0x7facb8d719
8d <vm_exec_core+7117>, 0x7facb8d71940 <vm_exec_core+7040>, 0x7facb8d71829 <vm_exec_core+6761>, 0x
7facb8d716bc <vm_exec_core+6396>, 0x7facb8d715e4 <vm_exec_core+6180>, 0x7facb8d714e3 <vm_exec_core
+5923>, 0x7facb8d72955 <vm_exec_core+11157>, 0x7facb8d7149a <vm_exec_core+5850>, 0x7facb8d71432 <v
m_exec_core+5746>, 0x7facb8d713ca <vm_exec_core+5642>, 0x7facb8d71386 <vm_exec_core+5574>, 0x7facb
8d712c2 <vm_exec_core+5378>, 0x7facb8d71237 <vm_exec_core+5239>, 0x7facb8d73978 <vm_exec_core+1528
8>, 0x7facb8d73c7c <vm_exec_core+16060>, 0x7facb8d73cea <vm_exec_core+16170>, 0x7facb8d730bc <vm_e
xec_core+13052>, 0x7facb8d73d8b <vm_exec_core+16331>, 0x7facb8d73b5c <vm_exec_core+15772>, 0x7facb
8d73bba <vm_exec_core+15866>, 0x7facb8d734ca <vm_exec_core+14090>, 0x7facb8d73530 <vm_exec_core+14
192>, 0x7facb8d73596 <vm_exec_core+14294>, 0x7facb8d73687 <vm_exec_core+14535>, 0x7facb8d735fc <vm
```

```
_exec_core+14396>, 0x7facb8d736ed <vm_exec_core+14637>, 0x7facb8d731f9 <vm_exec_core+13369>, 0x7fa
cb8d73254 <vm_exec_core+13460>, 0x7facb8d739e7 <vm_exec_core+15399>, 0x7facb8d7118a <vm_exec_core+
5066>, 0x7facb8d710de <vm_exec_core+4894>, 0x7facb8d73b0c <vm_exec_core+15692>, 0x7facb8d73793 <vm
_exec_core+14803>, 0x7facb8d737e3 <vm_exec_core+14883>, 0x7facb8d73895 <vm_exec_core+15061>, 0x7fa
cb8d73903 <vm_exec_core+15171>, 0x7facb8d73a79 <vm_exec_core+15545>, 0x7facb8d71065 <vm_exec_core+
4773>, 0x7facb8d70fdd <vm_exec_core+4637>, 0x7facb8d70f1d <vm_exec_core+4445>, 0x7facb8d70ece <vm_
exec_core+4366>, 0x7facb8d70e7b <vm_exec_core+4283>, 0x7facb8d70e2a <vm_exec_core+4202>, 0x7facb8d
70dd1 <vm_exec_core+4113>, 0x7facb8d70d94 <vm_exec_core+4052>, 0x7facb8d70d57 <vm_exec_core+3991>,
 0x7facb8d7018d <vm_exec_core+973>, 0x7facb8d70119 <vm_exec_core+857>, 0x7facb8d700a8 <vm_exec_cor
e+744>, 0x7facb8d70564 <vm_exec_core+1956>, 0x7facb8d704e2 <vm_exec_core+1826>, 0x7facb8d70463 <vm
_exec_core+1699>, 0x7facb8d703a0 <vm_exec_core+1504>, 0x7facb8d70647 <vm_exec_core+2183>, 0x7facb8
d705e8 <vm_exec_core+2088>, 0x7facb8d702cf <vm_exec_core+1295>, 0x7facb8d701b7 <vm_exec_core+1015>
, 0x7facb8d70b93 <vm_exec_core+3539>, 0x7facb8d70b46 <vm_exec_core+3462>, 0x7facb8d70ae0 <vm_exec_
core+3360>, 0x7facb8d70a6b <vm_exec_core+3243>, 0x7facb8d70a26 <vm_exec_core+3174>, 0x7facb8d709e6
 <vm_exec_core+3110>, 0x7facb8d709a9 <vm_exec_core+3049>, 0x7facb8d7096c <vm_exec_core+2988>, 0x7f
acb8d7092f <vm_exec_core+2927>, 0x7facb8d708c4 <vm_exec_core+2820>, 0x7facb8d70878 <vm_exec_core+2
744>, 0x7facb8d70829 <vm_exec_core+2665>, 0x7facb8d707c9 <vm_exec_core+2569>, 0x7facb8d7077b <vm_e
xec_core+2491>, 0x7facb8d706f4 <vm_exec_core+2356>, 0x7facb8d706b2 <vm_exec_core+2290>, 0x7facb8d7
289a <vm_exec_core+10970>, 0x7facb8d72838 <vm_exec_core+10872>, 0x7facb8d728ff <vm_exec_core+11071
>, 0x7facb8d720e9 <vm_exec_core+9001>, 0x7facb8d71ee0 <vm_exec_core+8480>, 0x7facb8d71e4a <vm_exec
_core+8330>, 0x7facb8d71de4 <vm_exec_core+8228>, 0x7facb8d71d84 <vm_exec_core+8132>, 0x7facb8d71d2
2 <vm_exec_core+8034>, 0x7facb8d71ccd <vm_exec_core+7949>, 0x7facb8d71c51 <vm_exec_core+7825>, 0x7
facb8d71bff <vm_exec_core+7743>, 0x7facb8d71bd0 <vm_exec_core+7696>, 0x7facb8d71b8d <vm_exec_core+
7629>, 0x7facb8d71b34 <vm_exec_core+7540>, 0x7facb8d71af8 <vm_exec_core+7480>, 0x7facb8d723ae <vm_
exec_core+9710>, 0x7facb8d72363 <vm_exec_core+9635>, 0x7facb8d72317 <vm_exec_core+9559>, 0x7facb8d
722dd <vm_exec_core+9501>, 0x7facb8d72266 <vm_exec_core+9382>, 0x7facb8d721fc <vm_exec_core+9276>,
 0x7facb8d721a5 <vm_exec_core+9189>, 0x7facb8d7212e <vm_exec_core+9070>, 0x7facb8d72610 <vm_exec_c
ore+10320>, 0x7facb8d724f5 <vm_exec_core+10037>, 0x7facb8d724b1 <vm_exec_core+9969>, 0x7facb8d7246
2 <vm_exec_core+9890>, 0x7facb8d72736 <vm_exec_core+10614>, 0x7facb8d72676 <vm_exec_core+10422>, 0
x7facb8d732df <vm_exec_core+13599>, 0x7facb8d727a8 <vm_exec_core+10728>, 0x7facb8d70cef <vm_exec_c
ore+3887>, 0x7facb8d71aa1 <vm_exec_core+7393>, 0x7facb8d71a4a <vm_exec_core+7306>, 0x7facb8d732f5
<vm_exec_core+13621>, 0x7facb8d719f3 <vm_exec_core+7219>, 0x7facb8d7197c <vm_exec_core+7100>, 0x7f
acb8d7192f <vm_exec_core+7023>, 0x7facb8d71818 <vm_exec_core+6744>, 0x7facb8d716ab <vm_exec_core+6
379>, 0x7facb8d715d3 <vm_exec_core+6163>, 0x7facb8d714d2 <vm_exec_core+5906>, 0x7facb8d72944 <vm_e
xec_core+11140>, 0x7facb8d71489 <vm_exec_core+5833>, 0x7facb8d71421 <vm_exec_core+5729>, 0x7facb8d
713b9 <vm_exec_core+5625>, 0x7facb8d71375 <vm_exec_core+5557>, 0x7facb8d712b1 <vm_exec_core+5361>,
 0x7facb8d71226 <vm_exec_core+5222>, 0x7facb8d7330b <vm_exec_core+13643>, 0x7facb8d73321 <vm_exec_
core+13665>, 0x7facb8d73337 <vm_exec_core+13687>, 0x7facb8d7334d <vm_exec_core+13709>, 0x7facb8d73
363 <vm_exec_core+13731>, 0x7facb8d73379 <vm_exec_core+13753>, 0x7facb8d7338f <vm_exec_core+13775>
, 0x7facb8d733ac <vm_exec_core+13804>, 0x7facb8d733c2 <vm_exec_core+13826>, 0x7facb8d733d8 <vm_exe
c_core+13848>, 0x7facb8d733ee <vm_exec_core+13870>, 0x7facb8d73404 <vm_exec_core+13892>, 0x7facb8d
7341a <vm_exec_core+13914>...}
#49 0x00007facb8d77039 in rb_vm_exec (ec=0x7fac803c97d0) at vm.c:2586
        _ec = 0x7fac803c97d0
        _tag = {tag = 36, retval = 4, buf = {0x0, 0x7facb8d77040 <rb_vm_exec+400>, 0x0, 0x7fac7191
7bb0, 0x0}, prev = 0x7fac71917ca0, state = RUBY_TAG_NONE, lock_rec = 0}
        state = RUBY_TAG_NONE
        result = <optimized out>
#50 0x00007fac8454d5e2 in ??? ()
#51 0x00007fac71918080 in ??? ()
#52 0x00007fac803c97d0 in ??? ()
#53 0x00007fac71937f58 in ??? ()
#54 0x00007facb8d76f98 in jit_exec_exception (ec=<optimized out>) at vm.c:509
        func = <optimized out>
        func = <optimized out>
#55 vm_exec_loop (result=<optimized out>, tag=<optimized out>, state=<optimized out>, ec=<optimize
d out>) at vm.c:2612
        vm_loop_start = <optimized out>
        vm_loop_start = <optimized out>
#56 rb_vm_exec (ec=0x7fac803c97d0) at vm.c:2589
        _ec = 0x7fac803c97d0
        _tag = {tag = 36, retval = 4, buf = {0x0, 0x7facb8d77040 <rb_vm_exec+400>, 0x0, 0x7fac7191
7c80, 0x0}, prev = 0x7fac71917d70, state = RUBY_TAG_NONE, lock_rec = 0}
        state = RUBY_TAG_NONE
        result = <optimized out>
#57 0x00007fac842cd7b8 in ??? ()
--Type <RET> for more, q to quit, c to continue without paging--
```

```
#58 0x00007fac71918048 in ??? ()
#59 0x00007fac803c97d0 in ??? ()
#60 0x00007fac71937f90 in ??? ()
#61 0x00007facb8d76f98 in jit_exec_exception (ec=<optimized out>) at vm.c:509
        func = <optimized out>
        func = <optimized out>
#62 vm_exec_loop (result=<optimized out>, tag=<optimized out>, state=<optimized out>, ec=<optimize
d out>) at vm.c:2612
        vm_loop_start = <optimized out>
        vm_loop_start = <optimized out>
#63 rb_vm_exec (ec=0x7fac803c97d0) at vm.c:2589
        _ec = 0x7fac803c97d0
        _tag = {tag = 36, retval = 4, buf = {0x0, 0x7facb8d77040 <rb_vm_exec+400>, 0x0, 0x7fac7191
7d50, 0x7fac9be125b0}, prev = 0x7fac71917e40, state = RUBY_TAG_NONE, lock_rec = 0}
        state = RUBY_TAG_NONE
        result = <optimized out>
#64 0x00007fac842cd6ce in ??? ()
#65 0x00007fac803c9780 in ??? ()
#66 0x0000000000000000 in ??? ()
```

Related issues:

Related to Ruby - Bug #21087: "try to mark T_NONE object" error in Rage/Activ...	**Closed**
Is duplicate of Ruby - Bug #21021: "try to mark T_NONE object" with 3.4.1	**Closed**

History

#1 - 01/14/2025 12:41 AM - travisbell (Travis Bell)

Could be related to the issue posted here: https://bugs.ruby-lang.org/issues/21021 but I am not using ActiveJob so hard to say if we're crossing the same paths or not.

#2 - 01/16/2025 12:09 AM - alanwu (Alan Wu)

Can you post the crash report ruby generates? By default it goes to stderr, and you can get it in a file by setting RUBY_CRASH_REPORT to a path if that's easier.

#3 - 01/16/2025 08:49 AM - Benoit_Tigeot (Benoit Tigeot)

travisbell (Travis Bell) wrote in #note-1:

> Could be related to the issue posted here: https://bugs.ruby-lang.org/issues/21021 but I am not using ActiveJob so hard to say if we're crossing the same paths or not.

By any chance could you share your WeakMap result like I did?

https://bugs.ruby-lang.org/issues/21021#note-10

#4 - 01/18/2025 12:12 AM - travisbell (Travis Bell)

Here's a crash dump: https://gist.github.com/travisbell/fc3b03a5d1bd797e16e990c17fb2ec9a

Based on what I am seeing here, I think it is the same crash as 21021 (do you agree?) If so, feel free to close this ticket, and we can concentrate on 21021.

#5 - 01/18/2025 12:38 AM - tenderlovemaking (Aaron Patterson)

These two lines make it seem like they are the same bug to me:

```
/usr/local/lib/libruby.so.3.4(gc_mark+0x16) [0x7f8e88955bae] gc/default/default.c:4456
/usr/local/lib/libruby.so.3.4(rb_gc_mark_vm_stack_values) gc/default/default.c:4436
```

@travisbell the backtrace from your corefile isn't enough to debug this. I can tell from this output:

```
#13 rb_gc_mark_vm_stack_values (n=62, values=0x7fac6ca72000) at gc.c:2291
        i = 29
```

that the 29th item (out of 62 items) in the VM stack has gone bad. Each Ruby stack frame in your program will have pushed some number of items on the Ruby stack before calling the next function. I can't tell from the gdb stack trace what functions push what, but it seems like maybe about halfway up your Ruby stack there's something wrong.

I think we can concentrate on #21021, but I'm not sure how much I can help without a core file or a repro program.

#6 - 01/18/2025 12:41 AM - tenderlovemaking (Aaron Patterson)

If you're able to get a gdb session, if you go to frame 13 (where rb_gc_mark_vm_stack_values is called) and try printing the objects that are on the stack near the bad address that might help clue us in to where the bug is originating. values should just be an array of Ruby objects, if we can print maybe values[i - 1] and values[i + 1] that might give us a clue.

#7 - 01/22/2025 12:24 AM - travisbell (Travis Bell)

Thanks Aaron.

This is a bit out of my wheelhouse. Here, I've got a new backtrace:

```
(gdb) bt
#0  __pthread_kill_implementation (no_tid=0, signo=6, threadid=<optimized out>) at ./nptl/pthread_kill.c:44
#1  __pthread_kill_internal (signo=6, threadid=<optimized out>) at ./nptl/pthread_kill.c:78
#2  __GI___pthread_kill (threadid=<optimized out>, signo=signo@entry=6) at ./nptl/pthread_kill.c:89
#3  0x00007fb63703926e in __GI_raise (sig=sig@entry=6) at ../sysdeps/posix/raise.c:26
#4  0x00007fb63701c8ff in __GI_abort () at ./stdlib/abort.c:79
#5  0x00007fb6373a38e4 in die () at error.c:1083
#6  rb_bug (fmt=fmt@entry=0x7fb637821abc "try to mark T_NONE object") at error.c:1117
#7  0x00007fb6374a6bae in gc_mark (objspace=0x7fb636872000, obj=140419388709720) at gc/default/default.c:4456
#8  gc_mark (obj=140419388709720, objspace=0x7fb636872000) at gc/default/default.c:4436
#9  gc_mark_and_pin (obj=140419388709720, objspace=0x7fb636872000) at gc/default/default.c:4482
#10 rb_gc_impl_mark_and_pin (obj=140419388709720, objspace_ptr=0x7fb636872000) at gc/default/default.c:4514
#11 gc_mark_and_pin_internal (obj=140419388709720) at gc.c:2259
#12 gc_mark_and_pin_internal (obj=140419388709720) at gc.c:2257
#13 rb_gc_mark_vm_stack_values (n=64, values=0x7fb5eea78000) at gc.c:2346
#14 0x00007fb63767ab49 in rb_execution_context_mark (ec=0x7fb5fefa9350) at vm.c:3415
#15 0x00007fb6374552bc in cont_mark (ptr=0x7fb5fefa9300) at cont.c:1019
#16 fiber_mark (ptr=0x7fb5fefa9300) at cont.c:1148
#17 0x00007fb6374ab67d in gc_mark_children (obj=140419656621520, objspace=0x7fb636872000) at gc/default/defaul
t.c:4640
#18 gc_mark_stacked_objects (incremental=1, count=4434, objspace=0x7fb636872000) at gc/default/default.c:4661
#19 gc_mark_stacked_objects_incremental (count=4434, objspace=0x7fb636872000) at gc/default/default.c:4693
#20 gc_marks_step (slots=4434, objspace=0x7fb636872000) at gc/default/default.c:5728
#21 gc_marks_continue (heap=0x7fb636872028, objspace=0x7fb636872000) at gc/default/default.c:5748
#22 gc_continue (objspace=0x7fb636872000, heap=0x7fb636872028) at gc/default/default.c:2073
#23 0x00007fb6374aee12 in newobj_cache_miss (objspace=objspace@entry=0x7fb636872000, cache=cache@entry=0x7fb63
687c1e0, heap_idx=heap_idx@entry=0, vm_locked=<optimized out>, vm_locked@entry=false) at gc/default/default.c:
2413
#24 0x00007fb6374b08e9 in newobj_alloc (vm_locked=false, heap_idx=0, cache=0x7fb63687c1e0, objspace=0x7fb63687
2000) at gc/default/default.c:2446
#25 rb_gc_impl_new_obj (alloc_size=40, wb_protected=true, v3=0, v2=0, v1=0, flags=8197, klass=140420580174920,
 cache_ptr=0x7fb63687c1e0, objspace_ptr=0x7fb636872000) at gc/default/default.c:2532
#26 newobj_of (size=40, wb_protected=true, v3=0, v2=0, v1=0, flags=8197, klass=140420580174920, cr=0x7fb63686d
280) at gc.c:1024
#27 rb_wb_protected_newobj_of (ec=<optimized out>, klass=klass@entry=140420580174920, flags=flags@entry=8197,
 size=size@entry=40) at gc.c:1062
#28 0x00007fb637611eb1 in str_alloc_heap (klass=140420580174920) at string.c:983
#29 str_new_shared (str=140420103491480, klass=140420580174920) at string.c:1451
#30 rb_sym_to_s (sym=<optimized out>) at string.c:12152
#31 0x00007fb6029c36b8 in ??? ()
#32 0x00007fb5f1e74cc0 in ??? ()
#33 0x00007fb5f1e76b98 in ??? ()
#34 0x00007fb63786ef80 in ??? () at /usr/local/lib/libruby.so.3.4
--Type <RET> for more, q to quit, c to continue without paging--c
#35 0x00007fb6376857e8 in jit_exec_exception (ec=<optimized out>) at vm.c:509
#36 vm_exec_loop (result=<optimized out>, tag=<optimized out>, state=<optimized out>, ec=<optimized out>) at v
m.c:2621
#37 rb_vm_exec (ec=0x7fb5fdfcd750) at vm.c:2598
#38 0x00007fb63768b2be in vm_call_bmethod_body (argv=0x7fb5f1e76b98, calling=0x7fb5f1e74cc0, ec=0x7fb5fdfcd750
) at /tmp/ruby-build.20250102181333.18.mVFuok/ruby-3.4.1/vm_insnhelper.c:4020
#39 vm_call0_body (ec=ec@entry=0x7fb5fdfcd750, calling=calling@entry=0x7fb5f1e74cc0, argv=0x7fb5f1e76b98) at /
tmp/ruby-build.20250102181333.18.mVFuok/ruby-3.4.1/vm_eval.c:244
#40 0x00007fb63768c6d5 in vm_call0_cc (kw_splat=0, cc=0x7fb5f1e74c60, argv=<optimized out>, argc=<optimized ou
t>, id=998143, recv=<optimized out>, ec=0x7fb5fdfcd750) at /tmp/ruby-build.20250102181333.18.mVFuok/ruby-3.4.1
/vm_eval.c:101
#41 rb_vm_call0 (kw_splat=0, cme=<optimized out>, argv=<optimized out>, argc=0, id=998143, recv=<optimized out
>, ec=0x7fb5fdfcd750) at /tmp/ruby-build.20250102181333.18.mVFuok/ruby-3.4.1/vm_eval.c:61
#42 rb_vm_call_kw (ec=ec@entry=0x7fb5fdfcd750, recv=<optimized out>, id=998143, argc=argc@entry=0, argv=argv@e
ntry=0x7fb5f1e76b98, me=<optimized out>, kw_splat=0) at /tmp/ruby-build.20250102181333.18.mVFuok/ruby-3.4.1/vm
_eval.c:326
#43 0x00007fb637571d6f in call_method_data (data=<optimized out>, kw_splat=0, passed_procval=<optimized out>,
 argv=0x7fb5f1e76b98, argc=0, ec=0x7fb5fdfcd750) at proc.c:2534
#44 rb_method_call_with_block_kw (argc=0, argv=0x7fb5f1e76b98, method=140419494254640, passed_procval=<optimiz
```

```
ed out>, kw_splat=0) at proc.c:2556
#45 0x00007fb602d038ee in ??? ()
#46 0x00007fb5f1e76b78 in ??? ()
#47 0x00007fb5fdfcd750 in ??? ()
#48 0x00007fb5f1e95350 in ??? ()
#49 0x00007fb6376857e8 in jit_exec_exception (ec=<optimized out>) at vm.c:509
#50 vm_exec_loop (result=<optimized out>, tag=<optimized out>, state=<optimized out>, ec=<optimized out>) at v
m.c:2621
#51 rb_vm_exec (ec=0x7fb5fdfcd750) at vm.c:2598
#52 0x00007fb602ea01a3 in ??? ()
#53 0x00007fb5f1e76b10 in ??? ()
#54 0x00007fb5fdfcd750 in ??? ()
#55 0x00007fb5f1e953f8 in ??? ()
#56 0x00007fb6376857e8 in jit_exec_exception (ec=<optimized out>) at vm.c:509
#57 vm_exec_loop (result=<optimized out>, tag=<optimized out>, state=<optimized out>, ec=<optimized out>) at v
m.c:2621
#58 rb_vm_exec (ec=0x7fb5fdfcd750) at vm.c:2598
#59 0x00007fb602d0371e in ??? ()
#60 0x00007fb5fdfcd750 in ??? ()
#61 0x0000000000000000 in ??? ()

(gdb) frame 13
#13 rb_gc_mark_vm_stack_values (n=64, values=0x7fb5eea78000) at gc.c:2346
warning: 2346 gc.c: No such file or directory

(gdb) print *values@n
$1 = {0, 0, 2040070307, 4, 4, 0, 140419256733801, 572653601, 140419659551200, 4, 0, 140419256731113, 572653601
, 0, 140419256731017, 572653601, 140419838444320, 140419655214160, 4, 140419983151800, 0, 286326787, 140419839
352160, 140420024205520, 140419983928960, 0, 140419656621240,
  140419839352160, 559836102393857, 140419659551200, 286326787, 140419388709720, 140419983199080, 140419369893
577, 286326787, 140419659551200, 140419656621280, 140419659550560, 4, 140419659550400, 140420097064960, 140419
369893577, 286326819, 140419659550400, 140419656621040, 140420097065080, 0,
  140419656620960, 140419659550400, 4, 140420097063760, 140419656620920, 140419656620880, 140419838483200, 140
419656621000, 140419656620800, 140420095167200, 0, 286326787, 140419983198880, 140419656621000, 14042009834068
0, 0, 1431634051}
```

How do get the values around the bad address?

#8 - 01/22/2025 11:54 PM - tenderlovemaking (Aaron Patterson)

@travisbell if you can do it, try to load this gdbinit script inside your gdb session (either copy the file to your home directory or the directory where you're running gdb). If you can load the gdbinit script, it should give you a command in gdb called rp (I think it stands for "Ruby print"). The rb command knows how to print out Ruby objects in a human friendly form.

In your case, first go to frame 13, then try to print the i variable. In your previous stack track, it showed that i is equal to 29, but it might be different in this core file, so double check it.

Once you've got the value for i try this in gdb: rp values[i + 1]. That *should* print out the object on one side of the bad entry. Also try with i - 1. We want to see if we can find objects in the stack, near the problem object, that will give us a clue about the code that is broken. You should also be able to do rp values[i] and hopefully it will display T_NONE (like the crash says).

The values you've printed above are all of the entries on your VM stack. Each entry is a Ruby object (or is a pointer masquerading as a Ruby object), so it should be safe to call rp on any of the addresses. So for example you could do something like rp 140419656620920.

In addition to Ruby objects, the VM stack should also point at 3 special values that correspond with every frame push. This value should correspond with the method (or block) being called.

For example if you have code like this that crashes inside the bar method:

```
def bar(x)
  # program crashes here before executing x + 1
  x + 1
end

def foo
  bar(123)
end

foo
```

Part of stack will look like this:

```
1. method entry for foo
2. block handler or previous ep pointer or 0 (for handling block locals, in this case 0 since it's a method)
3. flags indicating the type of code we're executing (in this case a method)
```

```
4. self
5. 123 (the value of x)
6. method entry for bar
7. same as 2
8. same as 3
```

Given this pattern, it should be possible to figure out which method pushed the bad value.

I'm really sorry this is so much work. I hope this explaination helps.

#9 - 01/23/2025 07:01 PM - travisbell (Travis Bell)

Awesome! Thanks for the how to. Really appreciate it.

Alright, here's the results;

```
(gdb) frame 13
#13 rb_gc_mark_vm_stack_values (n=64, values=0x7f26022c0000) at gc.c:2346
warning: 2346 gc.c: No such file or directory

(gdb) rp i
FIXNUM: 15

(gdb) rp values[i + 1]
A syntax error in expression, near `) & RUBY_FIXNUM_FLAG'.

(gdb) rp values[i - 1]
A syntax error in expression, near `) & RUBY_FIXNUM_FLAG'.

(gdb) rp values[i]
T_NONE: $1 = (struct RBasic *) 0x7f25fd324428
```

#10 - 01/23/2025 07:39 PM - alanwu (Alan Wu)

Curious if you get this crash without YJIT?

#11 - 01/23/2025 07:51 PM - travisbell (Travis Bell)

Hi Alan, I haven't tried to be honest. Let me deploy now with it disabled and I will let you know. It reliably happens within minutes so it's very easy to check.

#12 - 01/23/2025 09:19 PM - alanwu (Alan Wu)

- Related to Bug #21087: "try to mark T_NONE object" error in Rage/ActiveRecord/Fiber with 3.4.1 upgrade added

#13 - 01/23/2025 09:49 PM - travisbell (Travis Bell)

alanwu (Alan Wu) wrote in #note-10:

> Curious if you get this crash without YJIT?

I can confirm, if I disable YJIT, the crashes seem to stop happening.

#14 - 01/28/2025 02:41 AM - hsbt (Hiroshi SHIBATA)

- Related to Bug #21021: "try to mark T_NONE object" with 3.4.1 added

#15 - 01/28/2025 08:46 AM - Benoit_Tigeot (Benoit Tigeot)

Happy to see a gdb working with the crash

travisbell (Travis Bell) wrote in #note-9:

> `A syntax error in expression, near `) & RUBY_FIXNUM_FLAG'.`

It seems those error are more related to what is passed to rp ?
https://github.com/ruby/ruby/blob/50e34fd7683ff77fae8c822096c8bf5f3ca12402/.gdbinit#L11 . Could it be rp (values[i + 1]) ?

tenderlovemaking (Aaron Patterson) wrote in #note-8:

> The values you've printed above are all of the entries on your VM stack. Each entry is a Ruby object (or is a pointer masquerading as a Ruby object), so it should be safe to call rp on any of the addresses. So for example you could do something like rp 140419656620920.

Did you try to do it @travisbell ?

#16 - 01/30/2025 04:06 PM - alanwu (Alan Wu)

- Is duplicate of Bug #21021: "try to mark T_NONE object" with 3.4.1 added

#17 - 01/30/2025 04:06 PM - alanwu (Alan Wu)

- Related to deleted (Bug #21021: "try to mark T_NONE object" with 3.4.1)

#18 - 01/30/2025 04:07 PM - travisbell (Travis Bell)

Ha, ya, thanks Benoit. I'm not sure what that was about, but here's the real values (I think):

```
(gdb) rp i
FIXNUM: 15

(gdb) rp values[14]
FIXNUM: 70068055741204

(gdb) rp values[15]
FIXNUM: 286326800

(gdb) rp values[16]
[PROMOTED] T_CLASS: (struct RClass *) 0x7f7423711c38
You can't do that without a process to debug.
```

Is that the output we expect?

#19 - 01/30/2025 04:10 PM - alanwu (Alan Wu)

- Status changed from Open to Closed

Should be fixed by 58ccce60cf5f3268e7ef27942b75e78fe2d78e75. If you'd like to try out a patch for 3.4.1, take a look at https://bugs.ruby-lang.org/issues/21021#note-26.