Ruby - Bug #19316

YJIT crash in 3.2.0

01/06/2023 02:35 PM - jdashton (J Daniel Ashton)

Status:	Closed		
Priority:	Normal		
Assignee:	jit		
Target version:			
ruby -v:	ruby 3.2.0 (2022-12-25 revision a528908271) +YJIT [x86_64-darwin22]	**Backport:**	2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONE, 3.2: DONE

Description

When I check out this commit from GitHub, https://github.com/jdashton/aoc2022-ruby/tree/5702dac483cd6e95f7be35bfebaf9d4a654796d8 , and run the following command, RUBYOPT="-v --yjit" bin/rspec spec/aoc2022/puzzles/unstable_diffusion_spec.rb, Ruby crashes.
Crash Report log file attached.

Associated revisions

Revision aeddc19340c7116d48fac3080553fbb823857d16 - 01/10/2023 04:11 PM - alanwu (Alan Wu)

YJIT: Save PC and SP before calling leaf builtins (#7090)

Previously, we did not update cfp->sp before calling the C function of
ISEQs marked with Primitive.attr! "inline" (leaf builtins). This
caused the GC to miss temporary values on the stack in case the function
allocates and triggers a GC run. Right now, there is only a few leaf
builtins in numeric.rb on Integer methods such as Integer#~. Since
these methods only allocate when operating on big numbers, we missed
this issue.

Fix by saving PC and SP before calling the functions -- our usual
protocol for calling C functions that may allocate on the GC heap.

[Bug #19316]

Revision aeddc19340c7116d48fac3080553fbb823857d16 - 01/10/2023 04:11 PM - alanwu (Alan Wu)

YJIT: Save PC and SP before calling leaf builtins (#7090)

Previously, we did not update cfp->sp before calling the C function of
ISEQs marked with Primitive.attr! "inline" (leaf builtins). This
caused the GC to miss temporary values on the stack in case the function
allocates and triggers a GC run. Right now, there is only a few leaf
builtins in numeric.rb on Integer methods such as Integer#~. Since
these methods only allocate when operating on big numbers, we missed
this issue.

Fix by saving PC and SP before calling the functions -- our usual
protocol for calling C functions that may allocate on the GC heap.

[Bug #19316]

Revision aeddc193 - 01/10/2023 04:11 PM - alanwu (Alan Wu)

YJIT: Save PC and SP before calling leaf builtins (#7090)

Previously, we did not update cfp->sp before calling the C function of
ISEQs marked with Primitive.attr! "inline" (leaf builtins). This
caused the GC to miss temporary values on the stack in case the function
allocates and triggers a GC run. Right now, there is only a few leaf
builtins in numeric.rb on Integer methods such as Integer#~. Since
these methods only allocate when operating on big numbers, we missed
this issue.

Fix by saving PC and SP before calling the functions -- our usual
protocol for calling C functions that may allocate on the GC heap.

[Bug #19316]

Revision 1fb5eb5740d4c4f1fc34a4a50bc0482eac27b545 - 01/18/2023 09:56 AM - naruse (Yui NARUSE)

merge revision(s) aeddc19340c7116d48fac3080553fbb823857d16: [Backport #19316]

```
    YJIT: Save PC and SP before calling leaf builtins (#7090)

    Previously, we did not update `cfp->sp` before calling the C function of
    ISEQs marked with `Primitive.attr! "inline"` (leaf builtins). This
    caused the GC to miss temporary values on the stack in case the function
    allocates and triggers a GC run. Right now, there is only a few leaf
    builtins in numeric.rb on Integer methods such as `Integer#~`. Since
    these methods only allocate when operating on big numbers, we missed
    this issue.

    Fix by saving PC and SP before calling the functions -- our usual
    protocol for calling C functions that may allocate on the GC heap.

    [Bug #19316]
    ---
     test/ruby/test_yjit.rb | 16 ++++++++++++++++
     yjit/src/codegen.rs    |  4 ++++
     2 files changed, 20 insertions(+)
```

Revision 1fb5eb5740d4c4f1fc34a4a50bc0482eac27b545 - 01/18/2023 09:56 AM - naruse (Yui NARUSE)

merge revision(s) aeddc19340c7116d48fac3080553fbb823857d16: [Backport #19316]

```
    YJIT: Save PC and SP before calling leaf builtins (#7090)

    Previously, we did not update `cfp->sp` before calling the C function of
    ISEQs marked with `Primitive.attr! "inline"` (leaf builtins). This
    caused the GC to miss temporary values on the stack in case the function
    allocates and triggers a GC run. Right now, there is only a few leaf
    builtins in numeric.rb on Integer methods such as `Integer#~`. Since
    these methods only allocate when operating on big numbers, we missed
    this issue.

    Fix by saving PC and SP before calling the functions -- our usual
    protocol for calling C functions that may allocate on the GC heap.

    [Bug #19316]
    ---
     test/ruby/test_yjit.rb | 16 ++++++++++++++++
     yjit/src/codegen.rs    |  4 ++++
     2 files changed, 20 insertions(+)
```

Revision 1fb5eb57 - 01/18/2023 09:56 AM - naruse (Yui NARUSE)

merge revision(s) aeddc19340c7116d48fac3080553fbb823857d16: [Backport #19316]

```
    YJIT: Save PC and SP before calling leaf builtins (#7090)

    Previously, we did not update `cfp->sp` before calling the C function of
    ISEQs marked with `Primitive.attr! "inline"` (leaf builtins). This
    caused the GC to miss temporary values on the stack in case the function
    allocates and triggers a GC run. Right now, there is only a few leaf
    builtins in numeric.rb on Integer methods such as `Integer#~`. Since
    these methods only allocate when operating on big numbers, we missed
    this issue.

    Fix by saving PC and SP before calling the functions -- our usual
    protocol for calling C functions that may allocate on the GC heap.

    [Bug #19316]
    ---
     test/ruby/test_yjit.rb | 16 ++++++++++++++++
     yjit/src/codegen.rs    |  4 ++++
     2 files changed, 20 insertions(+)
```

Revision c660aaf439dcd609e4e23253372c8ec6d567ce10 - 03/21/2023 03:10 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) aeddc19340c7116d48fac3080553fbb823857d16: [Backport #19316]

```
    YJIT: Save PC and SP before calling leaf builtins (#7090)

    Previously, we did not update `cfp->sp` before calling the C function of
```

```
    ISEQs marked with `Primitive.attr! "inline"` (leaf builtins). This
    caused the GC to miss temporary values on the stack in case the function
    allocates and triggers a GC run. Right now, there is only a few leaf
    builtins in numeric.rb on Integer methods such as `Integer#~`. Since
    these methods only allocate when operating on big numbers, we missed
    this issue.

    Fix by saving PC and SP before calling the functions -- our usual
    protocol for calling C functions that may allocate on the GC heap.

    [Bug #19316]
    ---
     test/ruby/test_yjit.rb | 16 ++++++++++++++++
     yjit/src/codegen.rs    |  4 ++++
     2 files changed, 20 insertions(+)
```

Revision c660aaf439dcd609e4e23253372c8ec6d567ce10 - 03/21/2023 03:10 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) aeddc19340c7116d48fac3080553fbb823857d16: [Backport #19316]

```
    YJIT: Save PC and SP before calling leaf builtins (#7090)

    Previously, we did not update `cfp->sp` before calling the C function of
    ISEQs marked with `Primitive.attr! "inline"` (leaf builtins). This
    caused the GC to miss temporary values on the stack in case the function
    allocates and triggers a GC run. Right now, there is only a few leaf
    builtins in numeric.rb on Integer methods such as `Integer#~`. Since
    these methods only allocate when operating on big numbers, we missed
    this issue.

    Fix by saving PC and SP before calling the functions -- our usual
    protocol for calling C functions that may allocate on the GC heap.

    [Bug #19316]
    ---
     test/ruby/test_yjit.rb | 16 ++++++++++++++++
     yjit/src/codegen.rs    |  4 ++++
     2 files changed, 20 insertions(+)
```

Revision c660aaf4 - 03/21/2023 03:10 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) aeddc19340c7116d48fac3080553fbb823857d16: [Backport #19316]

```
    YJIT: Save PC and SP before calling leaf builtins (#7090)

    Previously, we did not update `cfp->sp` before calling the C function of
    ISEQs marked with `Primitive.attr! "inline"` (leaf builtins). This
    caused the GC to miss temporary values on the stack in case the function
    allocates and triggers a GC run. Right now, there is only a few leaf
    builtins in numeric.rb on Integer methods such as `Integer#~`. Since
    these methods only allocate when operating on big numbers, we missed
    this issue.

    Fix by saving PC and SP before calling the functions -- our usual
    protocol for calling C functions that may allocate on the GC heap.

    [Bug #19316]
    ---
     test/ruby/test_yjit.rb | 16 ++++++++++++++++
     yjit/src/codegen.rs    |  4 ++++
     2 files changed, 20 insertions(+)
```

Revision 82d763c94ad693a2af8086df8e0455b7de2d2ce3 - 03/21/2023 05:16 AM - nagachika (Tomoyuki Chikanaga)

Skip the test for [Bug #19316] for a while.

Revision 82d763c94ad693a2af8086df8e0455b7de2d2ce3 - 03/21/2023 05:16 AM - nagachika (Tomoyuki Chikanaga)

Skip the test for [Bug #19316] for a while.

Revision 82d763c9 - 03/21/2023 05:16 AM - nagachika (Tomoyuki Chikanaga)

Skip the test for [Bug #19316] for a while.

Revision 6ee749a52817fc463bbc2e93e5c3874a8c9aacf9 - 03/25/2023 12:50 AM - nagachika (Tomoyuki Chikanaga)

Revert "Skip the test for [Bug #19316] for a while."

This reverts commit 82d763c94ad693a2af8086df8e0455b7de2d2ce3,
and add exit: :any to assert_compile.

Co-authored-by: Alan Wu alansi.xingwu@shopify.com

Revision 6ee749a52817fc463bbc2e93e5c3874a8c9aacf9 - 03/25/2023 12:50 AM - nagachika (Tomoyuki Chikanaga)

Revert "Skip the test for [Bug #19316] for a while."

This reverts commit 82d763c94ad693a2af8086df8e0455b7de2d2ce3,
and add exit: :any to assert_compile.

Co-authored-by: Alan Wu alansi.xingwu@shopify.com

Revision 6ee749a5 - 03/25/2023 12:50 AM - nagachika (Tomoyuki Chikanaga)

Revert "Skip the test for [Bug #19316] for a while."

This reverts commit 82d763c94ad693a2af8086df8e0455b7de2d2ce3,
and add exit: :any to assert_compile.

Co-authored-by: Alan Wu alansi.xingwu@shopify.com

History

#1 - 01/09/2023 11:03 AM - noahgibbs (Noah Gibbs)

Looks like a SIGABRT, address 0x38. This isn't one I've seen yet.

#2 - 01/09/2023 11:26 PM - alanwu (Alan Wu)

Thank you for the report and for providing a reliable repro -- it makes diagnosing the problem that much easier!
We have a fix in the pipeline now.

#3 - 01/09/2023 11:29 PM - k0kubun (Takashi Kokubun)

- Backport changed from 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN to 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED, 3.2: REQUIRED

Updated the Backport field. Looking at the patch, we probably need to backport this to 3.1 as well (the same logic needs to be rewritten in C for https://github.com/ruby/ruby/blob/v3_1_0/yjit_codegen.c#L3638-L3661).

#4 - 01/10/2023 03:29 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

- Assignee set to jit

#5 - 01/10/2023 04:11 PM - alanwu (Alan Wu)

- Status changed from Assigned to Closed

Applied in changeset git|aeddc19340c7116d48fac3080553fbb823857d16.

YJIT: Save PC and SP before calling leaf builtins (#7090)

Previously, we did not update cfp->sp before calling the C function of
ISEQs marked with Primitive.attr! "inline" (leaf builtins). This
caused the GC to miss temporary values on the stack in case the function
allocates and triggers a GC run. Right now, there is only a few leaf
builtins in numeric.rb on Integer methods such as Integer#~. Since
these methods only allocate when operating on big numbers, we missed
this issue.

Fix by saving PC and SP before calling the functions -- our usual
protocol for calling C functions that may allocate on the GC heap.

[Bug #19316]

#6 - 01/10/2023 07:12 PM - alanwu (Alan Wu)

Here is a reproducer for 3.1.3:

```
def foo(_, a, b, c)
  a & b & ~c
end

n = 2 ** 64
args = [0, -n, n, n-1]

GC.stress = true
p foo(0, -n, n, n-1)
p foo(0, -n, n, n-1)
p foo(0, -n, n, n-1)

__END__
$ ruby-3.1.3/bin/ruby test.rb
18446744073709551616
18446744073709551616
18446744073709551616
$ ruby-3.1.3/bin/ruby --yjit-call-threshold=1 test.rb
18446744073709551616
18446744073709551616
-18446744073709551616
```

Patch:

```
diff --git a/yjit_codegen.c b/yjit_codegen.c
--- a/yjit_codegen.c
+++ b/yjit_codegen.c
@@ -3638,6 +3638,8 @@ gen_send_iseq(jitstate_t *jit, ctx_t *ctx, const struct rb_callinfo *ci, const r
     if (leaf_builtin && !block && leaf_builtin->argc + 1 <= NUM_C_ARG_REGS) {
         ADD_COMMENT(cb, "inlined leaf builtin");

+       jit_prepare_routine_call(jit, ctx, REG0);
+
         // Call the builtin func (ec, recv, arg1, arg2, ...)
         mov(cb, C_ARG_REGS[0], REG_EC);
```

It passes make check for me locally when applied against the release tar ball.

#7 - 01/20/2023 08:01 AM - naruse (Yui NARUSE)

- Backport changed from 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED, 3.2: REQUIRED to 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED, 3.2: DONE

ruby_3_2 1fb5eb5740d4c4f1fc34a4a50bc0482eac27b545 merged revision(s) aeddc19340c7116d48fac3080553fbb823857d16.

#8 - 03/21/2023 03:46 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED, 3.2: DONE to 2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONE, 3.2: DONE

ruby_3_1 c660aaf439dcd609e4e23253372c8ec6d567ce10 merged revision(s) aeddc19340c7116d48fac3080553fbb823857d16.

#9 - 03/21/2023 03:47 AM - nagachika (Tomoyuki Chikanaga)

@alanwu (Alan Wu) Thank you for providing the patch for ruby_3_1. I have applied it at c660aaf439dcd609e4e23253372c8ec6d567ce10.

#10 - 03/21/2023 05:08 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONE, 3.2: DONE to 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED, 3.2: DONE

make test-alll on GitHub Actions with YJIT fails as follows.

```
    1) Failure:
  TestYJIT#test_bug_19316 [/home/runner/work/ruby/ruby/src/test/ruby/test_yjit.rb:691]:
  Expected no exits, but got
  {:opt_send_without_block=>1, :leave=>2, :opt_and=>1}

  Finished tests in 438.457308s, 49.5715 tests/s, 6283.6517 assertions/s.
  21735 tests, 2755113 assertions, 1 failures, 0 errors, 101 skips
```

Should I apply some additional changes?
I restore the Backport field for 3.1 to "REQUIRED".

#11 - 03/21/2023 08:57 PM - alanwu (Alan Wu)

@nagachika (Tomoyuki Chikanaga) You shouldn't need any other code changes. To fix CI, pass exit: :any in the test:

```
   def test_bug_19316
-    omit "skip this test for [Bug #19316] for a while."
     n = 2 ** 64
     # foo's extra param and the splats are relevant
-    assert_compiles(<<~'RUBY', result: [[n, -n], [n, -n]])
+    assert_compiles(<<~'RUBY', result: [[n, -n], [n, -n]], exits: :any)
       def foo(_, a, b, c)
         [a & b, ~c]
       end
```

We only care about getting the correct result in this case so don't mind exiting from YJIT.
Keeping the omit is also probably fine.

#12 - 03/25/2023 01:19 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.7: DONTNEED, 3.0: DONTNEED, 3.1: REQUIRED, 3.2: DONE to 2.7: DONTNEED, 3.0: DONTNEED, 3.1: DONE, 3.2: DONE

@alanwu (Alan Wu) Thank you!
I applied the patch you provided at 6ee749a52817fc463bbc2e93e5c3874a8c9aacf9.

Files

ruby-2023-01-06-091855.ips	21.6 KB	01/06/2023	jdashton (J Daniel Ashton)