Ruby master - Bug #17488
Regression in Ruby 3: Hash#key? is non-deterministic when argument uses DelegateClass
12/28/2020 08:52 PM - myronmarston (Myron Marston)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin19]</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.5: REQUIRED, 2.6: REQUIRED, 2.7: REQUIRED, 3.0: DONE</td>
</tr>
</tbody>
</table>

Description
Upon upgrading a library to run on Ruby 3.0, I have observed that Hash#key? has non-deterministic behavior when the argument uses DelegateClass. This non-deterministic behavior was not present in Ruby 2.7.

Reproducing this is slightly difficult; the behavior appears to be deterministic (but not necessarily correct) within a single ruby process. To reproduce the non-determinism, you need to start ruby many times to observe different results. My script below does this.

Reproduction script

```
puts "Running on Ruby: #{RUBY_DESCRIPTION}"
program = <<~EOS
  require "delegate"
  TypeName = DelegateClass(String)

  hash = {
    "Int" => true,
    "Float" => true,
    "String" => true,
    "Boolean" => true,
    "WidgetFilter" => true,
    "WidgetAggregation" => true,
    "WidgetEdge" => true,
    "WidgetSortOrder" => true,
    "WidgetGrouping" => true,
  }

  puts hash.key?(TypeName.new("WidgetAggregation"))
EOS
iterations = 20
results = iterations.times.map { `ruby -e '#{program}'`.chomp }.tally
puts "Results of checking `Hash#key?' #{iterations} times: #{results.inspect}"
```

Put this in a file like ruby3_hash_bug.rb, and run it using either Ruby 2.7 (to see Hash#key? consistently return true) or Ruby 3.0 (to see Hash#key? produce non-deterministic behavior).

Ruby 2.7 results

Running on Ruby: ruby 2.7.1p83 (2020-03-31 revision a0c7c23c9c) [x86_64-darwin19]
Results of checking `Hash#key?' 20 times: {"true"=>20}

Ruby 3.0 results

Running on Ruby: ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin19]
Results of checking `Hash#key?' 20 times: {"true"=>12, "false"=>8}

Note that the ratio of true to false is non-deterministic; here are a couple other runs on Ruby 3.0 with different results:
Running on Ruby: ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin19]
Results of checking `Hash#key?` 20 times: {"false"=>7, "true"=>13}

Running on Ruby: ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin19]
Results of checking `Hash#key?` 20 times: {"true"=>11, "false"=>9}

Associated revisions
Revision 20a8425a - 12/31/2020 03:11 AM - nobu (Nobuyoshi Nakada)
Make any hash values fixable [Bug #17488]
As hnum is an unsigned st_index_t, the result of RSHIFT may not be in the fixable range.
Co-authored-by: NeoCat neocat@neocat.jp

Revision b2beb858 - 01/13/2021 08:06 AM - nobu (Nobuyoshi Nakada)
Make any hash values fixable [Bug #17488]
As hnum is an unsigned st_index_t, the result of RSHIFT may not be in the fixable range.
Co-authored-by: NeoCat neocat@neocat.jp

History
#1 - 12/28/2020 09:08 PM - marcandre (Marc-Andre Lafortune)
I didn't run a git bisect, but this was the case already in ruby 3.0.0preview1

#2 - 12/28/2020 09:12 PM - marcandre (Marc-Andre Lafortune)
Bug requires more than 8 keys (as in the example)

#3 - 12/29/2020 04:43 AM - nobu (Nobuyoshi Nakada)
Seems 9e6a39c35127a962c44dc3723bc96a0f8be90341 by bisect.

#4 - 12/30/2020 03:42 PM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Feedback
Hello, I cannot reproduce this on any of ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-linux] compiled using [clang-[3,4,5,6,7,8,9,10,11,12],gcc-[4,5,6,7,8,9,10]] on my Linux box. Maybe an XCode glitch I suspect?
I have just restored the ruby.h branch. nobu (Nobuyoshi Nakada) can you bisect on this branch as well to spot the actual change? Sorry for the trouble. It was my bad to press the squash button when I merged this.
https://github.com/shyouhei/ruby/tree/ruby.h
Also I want to know your clang --version.

#5 - 12/30/2020 04:14 PM - marcandre (Marc-Andre Lafortune)
shyouhei (Shyouhei Urabe) wrote in #note-4:
Also I want to know your clang --version.

$ clang --version
Apple clang version 11.0.0 (clang-1100.0.33.17)
Target: x86_64-apple-darwin18.7.0
Thread model: posix
InstalledDir: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin

#6 - 12/30/2020 04:18 PM - marcandre (Marc-Andre Lafortune)
On my mac pro (High Sierra) too:

$ clang --version
Apple LLVM version 10.0.0 (clang-1000.11.45.2)
Target: x86_64-apple-darwin17.7.0
Thread model: posix

03/15/2021
#7 - 12/30/2020 09:24 PM - sekiyama (Tomoki Sekiyama)
I have confirmed this too in ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-linux] built with gcc (Debian 8.3.0-6) 8.3.0.
And also confirmed that this PR fixes the issue:
https://github.com/ruby/ruby/pull/4014

#8 - 12/31/2020 03:13 AM - nobu (Nobuyoshi Nakada)
- Status changed from Feedback to Closed

Applied in changeset git|20a8425aa0f9a947e72b06cb3a2afe9674dd18f.

Make any hash values fixable [Bug #17488]

As hnum is an unsigned st_index_t, the result of RSHIFT may not be in the fixable range.

Co-authored-by: NeoCat neocat@neocat.jp

#9 - 12/31/2020 06:05 AM - nobu (Nobuyoshi Nakada)

shyouhei (Shyouhei Urabe) wrote in #note-4:

Hello, I cannot reproduce this on any of ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-linux] compiled using {clang-{3,4,5,6,7,8,10,11,12),gcc-{4,5,6,7,8,9,10}} on my Linux box. Maybe an XCode glitch I suspect?

I could confirm that test_any_hash_fixable fails on the followings.

gcc (Ubuntu 10.2.0-13ubuntu1) 10.2.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Apple clang version 11.0.3 (clang-1103.0.32.62)
Target: x86_64-apple-darwin19.6.0
Thread model: posix

InstalledDir: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin

gcc-mp-10 (MacPorts gcc10 10.2.0_4) 10.2.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

I have just restored the ruby.h branch. nobu (Nobuyoshi Nakada) can you bisect on this branch as well to spot the actual change? Sorry for the trouble. It was my bad to press the squash button when I merged this.

https://github.com/shyouhei/ruby/tree/ruby.h

03670392f46f00a4f9c20c3c7ea6215138037a6 is the first bad commit
commit 03670392f46f00a4f9c20c3c7ea6215138037a6
Author: 0000 <shyouhei@ruby-lang.org>
Date: Mon Mar 9 11:58:34 2020 +0900

- include/ruby/3/arithmetic/long.h rework
  - Turned macros into inline functions.

- include/ruby/3/arithmetic/long.h | 187 +++++++++++++++++++++-------------------
- include/ruby/3/attr/const.h | 11 +++
- include/ruby/3/attr/constexpr.h | 11 +++
- numeric.c | 2 --

4 files changed, 174 insertions(+), 37 deletions(-)

It also happens:

Assertion Failed: ./include/ruby/3/arithmetic/long.h:84:RB_INT2FIX:"{{(i) < (9223372036854775807L / 2) + 1) && (i) >= ((-9223372036854775807L -1L) / 2))}}"
It doesn't look the real cause though, just revealed.
The cause was that hnum <<= 1; hnum = RSHIFT(hnum, 1); just clears/sets the MSB, but not the next bit, so it can exceed Fixnum limits.
I'm not sure why it didn't happen before the commit.

#10 - 01/05/2021 02:35 AM - nobu (Nobuyoshi Nakada)
- Backport changed from 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN to 2.5: REQUIRED, 2.6: REQUIRED, 2.7: REQUIRED, 3.0: REQUIRED

#11 - 02/01/2021 09:22 AM - naruse (Yui NARUSE)
- Backport changed from 2.5: REQUIRED, 2.6: REQUIRED, 2.7: REQUIRED, 3.0: REQUIRED to 2.5: REQUIRED, 2.6: REQUIRED, 2.7: REQUIRED, 3.0: DONE

ruby_3_0 b2beb8586e930c168af434d6545f75d7612319b.

#12 - 02/20/2021 08:47 AM - nagachika (Tomoyuki Chikanaga)
MEMO: backporting 5f6053824551aec947a1c53d08975595aca1e513.20a8425aa0f9a947e72b06cb3a2afe9674dd18f but the test TestHash#test_any_hash_fixable and TestHash::TestSubHash#test_any_hash_fixable failed.

```
[11416/21074] TestHash#test_any_hash_fixable = 0.08 s
1) Failure:
TestHash#test_any_hash_fixable [//Users/nagachika/opt/ruby-2.7/src/ruby_2_7/test/ruby/test_hash.rb:1853]
Expected {
  "Int"=>true,
  "Float"=>true,
  "String"=>true,
  "Boolean"=>true,
  "WidgetFilter"=>true,
  "WidgetAggregation"=>true,
  "WidgetEdge"=>true,
  "WidgetSortOrder"=>true,
  "WidgetGrouping"=>true}.key?("Float") to return true.

[11552/21074] TestHash::TestSubHash#test_any_hash_fixable = 0.08 s
2) Failure:
TestHash::TestSubHash#test_any_hash_fixable [//Users/nagachika/opt/ruby-2.7/src/ruby_2_7/test/ruby/test_hash.rb:1853]
Expected {
  "Int"=>true,
  "Float"=>true,
  "String"=>true,
  "Boolean"=>true,
  "WidgetFilter"=>true,
  "WidgetAggregation"=>true,
  "WidgetEdge"=>true,
  "WidgetSortOrder"=>true,
  "WidgetGrouping"=>true}.key?("Int") to return true.
```