Ruby master - Feature #16635
Reduce allocation of begin-less/end-less range with Integer literal
02/15/2020 03:52 PM - pocke (Masataka Kuwabara)

Status: Closed
Priority: Normal
Assignee:
Target version:

Description
The patch is here. https://github.com/ruby/ruby/pull/2910

Currently Ruby optimizes Integer only Range with reducing Range allocation.
1..2 returns the same object.
For example:
$ ruby -e 'def a() 1..2 end; p a.object_id == a.object_id'
true

But it does not optimize begin-less/end-less ranges.
For example:
$ ruby -e 'def a() ..2 end; p a.object_id == a.object_id'
false
$ ruby -e 'def a() 1.. end; p a.object_id == a.object_id'
false

This patch applies the same optimization to begin-less/end-less ranges.

I think it will be helpful when we replace 1..-1 with ..-1 for String or Array slicing. Like str[1..-1]. We can rewrite str[1..-1] to str[1..] with keeping the speed with this patch.

Benchmark
It uses benchmark-driver gem.

# bench.yaml
prelude: |
  def endless
    1..
  end

  def beginless
    ..1
  end

  def endless_substr(str)
    str[1..]
  end

benchmark:
  endless: endless
  beginless: beginless
  endless_substr: "endless_substr('foo')"

$ benchmark-driver test.yaml --rbenv 'patched;trunk'
Warming up --------------------------------------
endless 45.948M 1/s - 46.076M times in 1.002782s (21.76ns/i, 26clocks/i)
beginless 49.986M 1/s - 50.237M times in 1.005037s (20.01ns/i, 24clocks/i)
endless_substr 8.067M 1/s - 8.187M times in 1.014936s (123.96ns/i, 148clocks/i)
Calculating -------------------------------------
patched  trunk
endless 115.679M 21.500M i/s - 137.843M times in 1.191597s 6.411398s
beginless 112.599M 22.060M i/s - 149.957M times in 1.331778s 6.797768s
endless_substr 8.888M 6.760M i/s - 24.201M times in 2.722995s 3.580038s

Comparison:

endless
patched: 115679391.9 i/s
trunk: 21499711.2 i/s - 5.38x slower

beginless
patched: 112598731.5 i/s
trunk: 22059673.0 i/s - 5.10x slower

endless_substr
patched: 8887513.1 i/s
trunk: 6759886.2 i/s - 1.31x slower

trunk is the HEAD of Ruby, and patched is trunk with this patch.

$ RBENV_VERSION=trunk ruby -v
ruby 2.8.0dev (2020-02-15T12:52:03Z master 961630126b) [x86_64-linux]
$ RBENV_VERSION=patched ruby -v
ruby 2.8.0dev (2020-02-15T12:52:03Z origin/master 961630126b) [x86_64-linux]

Associated revisions
Revision 52782942 - 02/15/2020 05:37 PM - pocke (Masataka Kuwabara)
Reduce begin-less/end-less range allocation

$ cat test.yaml
prelude: |
def endless
1..
end
def beginless
..1
end
def endless_substr(str)
str[1..]
end

benchmark:
endless: endless
beginless: beginless
endless_substr: "endless_substr('foo')"

$ RBENV_VERSION=trunk ruby -v
ruby 2.8.0dev (2020-02-15T12:52:03Z master 961630126b) [x86_64-linux]
$ RBENV_VERSION=patched ruby -v
ruby 2.8.0dev (2020-02-15T12:52:03Z origin/master 961630126b) [x86_64-linux]

$ benchmark-driver test.yaml --rbenv 'patched;trunk'
Warming up -----------------------------
endless 45.948M i/s - 46.076M times in 1.002782s (21.76ns/i, 26clocks/i)
beginless 49.986M i/s - 50.237M times in 1.005037s (20.01ns/i, 24clocks/i)
endless_substr 8.067M i/s - 8.187M times in 1.014936s (123.96ns/i, 148clocks/i)
Calculating -----------------------------
patched          trunk
endless 115.679M 21.500M i/s - 137.843M times in 1.191597s 6.411398s
beginless 112.599M 22.060M i/s - 149.957M times in 1.331778s 6.797768s
endless_substr 8.888M 6.760M i/s - 24.201M times in 2.722995s 3.580038s

Comparison:
endless
patched: 115679391.9 i/s
trunk: 21499711.2 i/s - 5.38x slower

beginless
patched: 112598731.5 i/s
History

#1 - 02/15/2020 04:05 PM - mame (Yusuke Endoh)
- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN)
- ruby -v deleted (ruby 2.8.0dev (2020-02-15T12:52:03Z master 961630126b) [x86_64-linux])
- Tracker changed from Bug to Feature

Looks good. I will merge after it passes the Cls. Thanks.

#2 - 02/18/2020 01:32 AM - k0kubun (Takashi Kokubun)
- Status changed from Open to Closed

Merged in 527829423088f09cf2f708be12bb4337d640dc69