Ruby - Feature #16291

Introduce support for resize in rb_ary_freeze and prefer internal use of rb_ary_freeze and rb_str_freeze for String and Array types

11/04/2019 01:45 PM - methodmissing (Lourens Naudé)

Status:	Assigned
Priority:	Normal
Assignee:	nobu (Nobuyoshi Nakada)
Target version:	

Description

References Github PR https://github.com/ruby/ruby/pull/2640

Why?

While working on https://github.com/ruby/ruby/pull/2037#issuecomment-548633141 I also looked at the rb_ary_freeze helper to determine if the same optimization of shrinking capacity can be applied there as well.

Further investigation revealed that rb_str_freeze attempts to resize / shrink strings on freeze, but we currently don't do the same for arrays despite API for it being exposed.

The gist of the change:

- Let rb_ary_freeze also attempt to right size an array before freezing it
- Replaced internal use of rb_obj_freeze and OBJ_FREEZE of callsites that pass a String or Array type in with rb_ary_freeze and rb_str_freeze specifically.

Results

Saves about 3MB of Array and String specific heap footprints on a redmine production env boot on Rails 6 (custom local upgrade - it does not officially support it yet):

```
Loading production environment (Rails 6.1.0.alpha)
irb(main):001:0> RUBY_DESCRIPTION
=> "ruby 2.7.0dev (2019-11-02T06:32:49Z master 772b0613c5) [x86_64-linux]"
irb(main):002:0> require 'objspace'
=> true
irb(main):003:0> GC.start
=> nil
irb(main):004:0> ObjectSpace.each_object(Array).sum {|o| ObjectSpace.memsize_of(o) }
=> 4451200
irb(main):005:0> ObjectSpace.each_object(String).sum {|o| ObjectSpace.memsize_of(o) }
=> 8608472
irb(main):006:0> exit
```

```
Loading production environment (Rails 6.1.0.alpha)
irb(main):001:0> RUBY_DESCRIPTION
=> "ruby 2.7.0dev (2019-11-02T16:52:34Z obj-freeze-specifi.. 3bc4048899) [x86_64-linux]"
irb(main):002:0> require 'objspace'
=> true
irb(main):003:0> GC.start
=> nil
irb(main):004:0> ObjectSpace.each_object(Array).sum {|o| ObjectSpace.memsize_of(o) }
=> 3233432
irb(main):005:0> ObjectSpace.each_object(String).sum {|o| ObjectSpace.memsize_of(o) }
=> 6748422
```

irb - the best calculator ...

```
irb(main):002:0> (4451200 - 3233432) / 1024
=> 1189
irb(main):003:0> (8608472 - 6748422) / 1024
=> 1816
```

Open questions

- Would it make sense to let rb_obj_freeze switch on object type instead and apply this optimization without "polluting" internals with custom callsite changes? Native extensions can then benefit from the resize feature too.
- HOWEVER: Discovered in testing that rb_str_freeze can fail asserts in rb_str_tmp_frozen_release when sprinkled too generously especially in io.c. And other issues may lurk too, thus incorporating the resizing in rb_obj_freeze won't work. See https://github.com/ruby/ruby/pull/2640#issuecomment-549119359

I think it could also make sense to split the PR in 2:

- One changeset for rb_ary_freeze attempting to resize Arrays on freeze, special case internal rb_obj_freeze of Arrays to prefer the new API
- Apply rb_str_resize at internal call sites where rb_obj_freeze is used with String types

Thoughts?

Related issues:

Related to Ruby - Feature #20589: Resize array in `rb_ary_freeze` and use `rb...	**Closed**

History

#1 - 11/05/2019 01:49 AM - shyouhei (Shyouhei Urabe)

Years ago I thought we don't need any per-class extension of rb_obj_freeze. However it seems rb_str_freeze is a thing and rb_ary_freeze can benefit the same way. I think it's now clear that there should be a C API that does rb_funcall(obj, rb_intern("freeze"), 0).

#2 - 11/08/2019 10:31 PM - methodmissing (Lourens Naudé)

Results of Array#frozen -> rb_ary_freeze with shrink support as per suggestion from Shyouhei:

```
lourens@CarbonX1:~/src/redmine$ irb
irb(main):001:0> RUBY_DESCRIPTION
=> "ruby 2.7.0dev (2019-11-05T10:33:43Z obj-freeze-specifi.. 485e4f3a84) [x86_64-linux]"
irb(main):002:0> require 'objspace'
=> true
irb(main):003:0> a = (1..5).to_a
irb(main):004:0> ObjectSpace.memsize_of(a)
=> 200
irb(main):005:0> a.freeze
=> [1, 2, 3, 4, 5]
irb(main):006:0> ObjectSpace.memsize_of(a)
=> 80
```

Master:

```
lourens@CarbonX1:~/src/redmine$ irb
irb(main):001:0> RUBY_DESCRIPTION
=> "ruby 2.7.0dev (2019-11-03T14:20:01Z master 5a7487bdcd) [x86_64-linux]"
irb(main):002:0> require 'objspace'
=> true
irb(main):003:0> a = (1..5).to_a
irb(main):004:0> ObjectSpace.memsize_of(a)
=> 200
irb(main):005:0> a.freeze
=> [1, 2, 3, 4, 5]
irb(main):006:0> ObjectSpace.memsize_of(a)
=> 200
```

#3 - 12/26/2019 02:24 AM - mame (Yusuke Endoh)

- Tracker changed from Misc to Feature

- Assignee set to nobu (Nobuyoshi Nakada)

- Target version set to 36

This ticket was discussed at the previous dev meeting, and @nobu (Nobuyoshi Nakada) will review the patch and merge it if okay.

#4 - 12/31/2019 06:43 AM - sam.saffron (Sam Saffron)

related, see this for for the string api

https://bugs.ruby-lang.org/issues/16029#change-82018

#5 - 09/29/2020 03:37 AM - hsbt (Hiroshi SHIBATA)

- Target version changed from 36 to 3.0

#6 - 12/25/2020 02:43 AM - naruse (Yui NARUSE)

- Target version deleted (3.0)

#7 - 04/03/2024 03:50 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

#8 - 06/20/2024 05:35 PM - byroot (Jean Boussier)

- Related to Feature #20589: Resize array in `rb_ary_freeze` and use `rb_ary_freeze` internally for arrays added