

Ruby - Bug #8211

Performance regression of method calls

04/03/2013 05:45 PM - dunric (David Unric)

Status:Closed Priority:Normal Assignee: Target version: ruby -v:2.0.0p0	Backport:
Description There are significant performance regressions in the newer Ruby versions. When compared to the last 1.9 patch build, method & lambda calls are about 3 times slower ! Benchmarking results: Ruby 1.9.3p392 user system total real lambda: 8.310000 0.000000 8.310000 (8.315247) instance method: 5.210000 0.000000 5.210000 (5.219299) class method: 5.040000 0.000000 5.040000 (5.037483) Ruby 2.0.0-p0 user system total real lambda: 25.310000 0.010000 25.320000 (25.339897) instance method: 17.870000 0.000000 17.870000 (17.882656) class method: 17.630000 0.010000 17.640000 (17.650515) Ruby 2.1.0-dev user system total real lambda: 19.210000 0.000000 19.210000 (19.227314) instance method: 13.930000 0.000000 13.930000 (13.936974) class method: 14.000000 0.000000 14.000000 (14.001582) The code: require 'benchmark' do_custom_stuff = lambda { 1 + 1 + 1 } class CustomStuff def self.do 1 + 1 + 1 end def do 1 + 1 + 1 end end custom_stuff = CustomStuff.new Benchmark.bm { b MANY = 50_000_000 b.report("lambda:") { MANY.times { do_custom_stuff.call } } b.report("instance method:") { MANY.times { custom_stuff.do } } b.report("class method:") { MANY.times { CustomStuff.do } } }	

History

#1 - 04/03/2013 06:07 PM - dunric (David Unric)

I had to add the testing platform is Linux x86_64 and interpreters were built with GCC 4.7.2 .

#2 - 04/04/2013 06:14 AM - drbrain (Eric Hodel)

- Priority changed from 5 to Normal

=begin

I can't reproduce with clang on OS X:

ruby 2.1.0dev (2013-04-04 trunk 40091) [x86_64-darwin12.3.0]

```
lambda:      7.310000  0.000000  7.310000 ( 7.303881)
instance method: 4.310000  0.000000  4.310000 ( 4.310044)
class method:   4.320000  0.000000  4.320000 ( 4.317902)
```

ruby 2.0.0p57 (2013-03-12 revision 39728) [x86_64-darwin12.2.1]

```
lambda:      7.290000  0.000000  7.290000 ( 7.290216)
instance method: 4.330000  0.000000  4.330000 ( 4.332266)
class method:   4.410000  0.000000  4.410000 ( 4.410439)
```

ruby 1.9.3p374 (2013-01-15 revision 38858) [x86_64-darwin12.2.1]

```
lambda:      8.010000  0.000000  8.010000 ( 8.014676)
instance method: 4.580000  0.000000  4.580000 ( 4.573837)
class method:   4.480000  0.000000  4.480000 ( 4.485607)
```

=end

#3 - 04/04/2013 07:33 AM - jonforums (Jon Forums)

=begin

I can't replicate the 1.9.3 vs. 2.0.0 results on my old Win7 32-bit box. 1.9.3 was built with mingw 4.6.2, and 2.0.0 was built with mingw-w64 4.7.2

==== ruby 1.9.3p408 (2013-04-02 revision 40056) [i386-mingw32]

```
lambda:      11.576000  0.000000  11.576000 ( 11.580016)
instance method: 7.456000  0.000000  7.456000 ( 7.460011)
class method:   7.676000  0.000000  7.676000 ( 7.670011)
```

==== ruby 2.0.0p102 (2013-04-03 revision 40075) [i386-mingw32]

```
lambda:      11.669000  0.000000  11.669000 ( 11.710016)
instance method: 7.051000  0.000000  7.051000 ( 7.050010)
class method:   6.864000  0.000000  6.864000 ( 6.880010)
```

=end

#4 - 04/04/2013 02:08 PM - bitsweat (Jeremy Daer)

David, this looks a lot like the CFLAGS issues folks were hitting with rvm. Essentially, getting unoptimized Ruby builds.

Take a close look at the differences in RbConfig::CONFIG between your Ruby builds.

```
ruby -e 'puts RUBY_DESCRIPTION, RbConfig::CONFIG["cflags"], RbConfig::CONFIG["CFLAGS"]'
```

How did you build Ruby?

#5 - 04/04/2013 04:53 PM - funny_falcon (Yura Sokolov)

Is your platform 32bit? Than it is cause of -fPIE

2013/4/4 bitsweat (Jeremy Kemper) jeremy@bitsweat.net

Issue [#8211](#) has been updated by bitsweat (Jeremy Kemper).

David, this looks a lot like the CFLAGS issues folks were hitting with rvm. Essentially, getting unoptimized Ruby builds.

Take a close look at the differences in RbConfig::CONFIG between your Ruby builds.

```
ruby -e 'puts RUBY_DESCRIPTION, RbConfig::CONFIG["cflags"],
RbConfig::CONFIG["CFLAGS"]'
```

How did you build Ruby?

Bug [#8211](https://bugs.ruby-lang.org/issues/8211#change-38202): Performance regression of method calls
<https://bugs.ruby-lang.org/issues/8211#change-38202>

Author: dunric (David Unric)
Status: Open
Priority: Normal
Assignee:
Category: core
Target version:
ruby -v: 2.0.0p0

There are significant performance regressions in the newer Ruby versions.
When compared to the last 1.9 patch build, method & lambda calls are about
3 times slower !

Benchmarking results:

Ruby 1.9.3p392

	user	system	total	real
lambda:	8.310000	0.000000	8.310000	(8.315247)
instance method:	5.210000	0.000000	5.210000	(5.219299)
class method:	5.040000	0.000000	5.040000	(5.037483)

Ruby 2.0.0-p0

	user	system	total	real
lambda:	25.310000	0.010000	25.320000	(25.339897)
instance method:	17.870000	0.000000	17.870000	(17.882656)
class method:	17.630000	0.010000	17.640000	(17.650515)

Ruby 2.1.0-dev

	user	system	total	real
lambda:	19.210000	0.000000	19.210000	(19.227314)
instance method:	13.930000	0.000000	13.930000	(13.936974)
class method:	14.000000	0.000000	14.000000	(14.001582)

The code:

```
require 'benchmark'

do_custom_stuff = lambda { 1 + 1 + 1 }

class CustomStuff
  def self.do
    1 + 1 + 1
  end

  def do
    1 + 1 + 1
  end
end

custom_stuff = CustomStuff.new

Benchmark.bm { |b|
  MANY = 50_000_000
  b.report("lambda:") { MANY.times { do_custom_stuff.call } }
  b.report("instance method:") { MANY.times { custom_stuff.do } }
  b.report("class method:") { MANY.times { CustomStuff.do } }
}
```

--
<http://bugs.ruby-lang.org/>

#6 - 04/05/2013 12:22 AM - dunric (David Unric)

bitsweat (Jeremy Kemper) wrote:

David, this looks a lot like the CFLAGS issues folks were hitting with rvm. Essentially, getting unoptimized Ruby builds.

Take a close look at the differences in RbConfig::CONFIG between your Ruby builds.

```
ruby -e 'puts RUBY_DESCRIPTION, RbConfig::CONFIG["cflags"], RbConfig::CONFIG["CFLAGS"]'
```

How did you build Ruby?

```
ruby -e 'puts RUBY_DESCRIPTION, RbConfig::CONFIG["cflags"], RbConfig::CONFIG["CFLAGS"]'
```

```
ruby 2.0.0p0 (2013-02-24 revision 39474) [x86_64-linux]
```

```
-O3 -fno-fast-math -ggdb3 -Wall -Wextra -Wno-unused-parameter -Wno-parentheses -Wno-long-long -Wno-missing-field-initializers  
-Wunused-variable -Werror=pointer-arith -Werror=write-strings -Werror=declaration-after-statement -Werror=implicit-function-declaration
```

#7 - 04/05/2013 12:39 AM - dunric (David Unric)

Asking to close this issue (reason: suboptimal compiler flags).

I've rebuilt Ruby with the following options and now get the expected performance:

```
ruby 2.0.0p0 (2013-02-24 revision 39474) [x86_64-linux]
```

```
-O3 -fno-fast-math -ggdb3 -Wall -Wextra -Wno-unused-parameter -Wno-parentheses -Wno-long-long -Wno-missing-field-initializers  
-Wunused-variable -Werror=pointer-arith -Werror=write-strings -Werror=declaration-after-statement -Werror=implicit-function-declaration  
-march=x86-64 -mtune=generic -O2 -pipe -fstack-protector --param=ssp-buffer-size=4 -D_FORTIFY_SOURCE=2 -fPIC
```

	user	system	total	real
--	------	--------	-------	------

lambda:	7.650000	0.000000	7.650000	(7.651665)
---------	----------	----------	----------	-------------

instance method:	4.350000	0.000000	4.350000	(4.352077)
------------------	----------	----------	----------	-------------

class method:	4.290000	0.000000	4.290000	(4.296082)
---------------	----------	----------	----------	-------------

#8 - 04/05/2013 04:53 AM - marcandre (Marc-Andre Lafortune)

- Status changed from Open to Closed