Ruby master - Bug #13447

Improve performance of rb_eql()

04/18/2017 07:50 AM - watson1978 (Shizuo Fujita)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:		**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

This is similar with https://github.com/ruby/ruby/pull/1552
At least, Array#eql? will be faster around 30% with following test code.

Before

```
       user     system      total        real
   1.740000   0.000000   1.740000 (  1.738344)
```

After

```
       user     system      total        real
   1.300000   0.000000   1.300000 (  1.303624)
```

Test code

```
require 'benchmark'

Benchmark.bmbm do |x|
  ary1 = Array.new(1000) { rand(1000) }
  ary2 = Array.new(1000) { rand(1000) }

  x.report do
    5000000.times do
      ary1.eql?(ary2)
    end
  end

end
```

Patch

https://github.com/ruby/ruby/pull/1589

Associated revisions

Revision b827fdff - 05/25/2017 04:25 AM - watson1978 (Shizuo Fujita)

Improve performance of rb_eql()

This improvement is similar with https://github.com/ruby/ruby/pull/1552

internal.h: add declaration of rb_eql_opt() API.

vm_insnhelper.c (rb_eql_opt): add rb_eql_opt() API which provides optimized
path for #eql? method such as rb_equal_opt().

object.c (rb_eql): optimize using rb_eql_opt() such as rb_equal().
Array#eql? and some methods have used rb_eql() and Array#eql? will be faster
around 20%.

```
[ruby-core:80761] [Bug #13447] [Fix GH-#1589]
```

Before

```
   user     system      total        real
```

1.570000   0.000000   1.570000 (1.569754)

After

```
   user     system      total        real
```

1.300000   0.000000   1.300000 (1.303624)

Test code

require 'benchmark'

Benchmark.bmbm do |x|
ary1 = Array.new(1000) { rand(1000) }
ary2 = Array.new(1000) { rand(1000) }

x.report do
5000000.times do
ary1.eql?(ary2)
end
end

end

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@58881 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 58881 - 05/25/2017 04:25 AM - watson1978 (Shizuo Fujita)

Improve performance of rb_eql()

This improvement is similar with https://github.com/ruby/ruby/pull/1552

internal.h: add declaration of rb_eql_opt() API.

vm_insnhelper.c (rb_eql_opt): add rb_eql_opt() API which provides optimized
path for #eql? method such as rb_equal_opt().

object.c (rb_eql): optimize using rb_eql_opt() such as rb_equal().
Array#eql? and some methods have used rb_eql() and Array#eql? will be faster
around 20%.

```
[ruby-core:80761] [Bug #13447] [Fix GH-#1589]
```

Before

```
   user     system      total        real
```

1.570000   0.000000   1.570000 (1.569754)

After

```
   user     system      total        real
```

1.300000   0.000000   1.300000 (1.303624)

Test code

require 'benchmark'

Benchmark.bmbm do |x|
ary1 = Array.new(1000) { rand(1000) }
ary2 = Array.new(1000) { rand(1000) }

x.report do
5000000.times do
ary1.eql?(ary2)
end
end

end

Revision 58881 - 05/25/2017 04:25 AM - watson1978 (Shizuo Fujita)

Improve performance of rb_eql()

This improvement is similar with https://github.com/ruby/ruby/pull/1552

internal.h: add declaration of rb_eql_opt() API.

vm_insnhelper.c (rb_eql_opt): add rb_eql_opt() API which provides optimized path for #eql? method such as rb_equal_opt().

object.c (rb_eql): optimize using rb_eql_opt() such as rb_equal().
Array#eql? and some methods have used rb_eql() and Array#eql? will be faster around 20%.

`[ruby-core:80761] [Bug #13447] [Fix GH-#1589]`

Before

```
   user     system      total        real
```

1.570000   0.000000   1.570000 (1.569754)

After

```
   user     system      total        real
```

1.300000   0.000000   1.300000 (1.303624)

Test code

```
require 'benchmark'

Benchmark.bmbm do |x|
ary1 = Array.new(1000) { rand(1000) }
ary2 = Array.new(1000) { rand(1000) }

x.report do
5000000.times do
ary1.eql?(ary2)
end
end

end
```

Revision 58881 - 05/25/2017 04:25 AM - watson1978 (Shizuo Fujita)

Improve performance of rb_eql()

This improvement is similar with https://github.com/ruby/ruby/pull/1552

internal.h: add declaration of rb_eql_opt() API.

vm_insnhelper.c (rb_eql_opt): add rb_eql_opt() API which provides optimized path for #eql? method such as rb_equal_opt().

object.c (rb_eql): optimize using rb_eql_opt() such as rb_equal().
Array#eql? and some methods have used rb_eql() and Array#eql? will be faster around 20%.

`[ruby-core:80761] [Bug #13447] [Fix GH-#1589]`

Before

```
   user     system      total        real
```

1.570000   0.000000   1.570000 (1.569754)

After

```
   user     system      total        real
```

1.300000   0.000000   1.300000 (1.303624)

Test code

```
require 'benchmark'

Benchmark.bmbm do |x|
ary1 = Array.new(1000) { rand(1000) }
ary2 = Array.new(1000) { rand(1000) }

x.report do
5000000.times do
ary1.eql?(ary2)
end
end

end
```

History

#1 - 04/18/2017 07:16 PM - nobu (Nobuyoshi Nakada)

Would you commit it by yourself?

#2 - 04/20/2017 05:56 AM - watson1978 (Shizuo Fujita)

Unfortunately, I'm just one of Ruby users.
Could you please commit it if you agreed the changing ?

#3 - 05/25/2017 04:25 AM - watson1978 (Shizuo Fujita)

- Status changed from Open to Closed

Applied in changeset trunk|r58881.

Improve performance of rb_eql()

This improvement is similar with https://github.com/ruby/ruby/pull/1552

internal.h: add declaration of rb_eql_opt() API.

vm_insnhelper.c (rb_eql_opt): add rb_eql_opt() API which provides optimized
path for #eql? method such as rb_equal_opt().

object.c (rb_eql): optimize using rb_eql_opt() such as rb_equal().
Array#eql? and some methods have used rb_eql() and Array#eql? will be faster
around 20%.

```
[ruby-core:80761] [Bug #13447] [Fix GH-#1589]
```

Before

```
   user     system      total        real
```

1.570000   0.000000   1.570000 (1.569754)

After

```
   user     system      total        real
```

1.300000   0.000000   1.300000 (1.303624)

Test code

```
require 'benchmark'

Benchmark.bmbm do |x|
ary1 = Array.new(1000) { rand(1000) }
ary2 = Array.new(1000) { rand(1000) }

x.report do
5000000.times do
ary1.eql?(ary2)
end
end
```

```
end
```