

Ruby master - Feature #12996

Optimize Range#===

12/02/2016 05:09 AM - ksss (Yuki Kurihara)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description The proposal is that Range#=== optimize by reducing method calls.	
Benchmark <pre>\$ cat t.rb i = 0 while i < 1_000_000 case i when 1..1000 end i += 1 end \$ time ./miniruby -e 1 ./miniruby -e 1 0.01s user 0.00s system 85% cpu 0.013 total</pre>	
Before <pre>\$ time ./miniruby t.rb ./miniruby t.rb 0.60s user 0.00s system 99% cpu 0.605 total</pre>	
After <pre>\$ time ./miniruby t.rb ./miniruby t.rb 0.41s user 0.00s system 99% cpu 0.420 total</pre>	
Important point Break compatibility in this case. https://github.com/ruby/ruby/blob/80045bc2a9e12012d6d16517ea5cb037e67eb2c1/test/ruby/test_range.rb#L333-L355 I don't know how to fix this issue. But I think, This is a key sentence maybe. https://github.com/ruby/ruby/blob/8130ee5c9dea6d1323d41271cc01c8dc5d8bcc5d/range.c#L1176	
Related issues:	
Related to Ruby master - Feature #12612: Switch Range#=== to use cover? inste...	Rejected
Related to Ruby master - Feature #14575: Switch Range#=== to use cover? inste...	Closed

History

#1 - 12/02/2016 11:20 AM - nobu (Nobuyoshi Nakada)

Yuki Kurihara wrote:

Important point

Break compatibility in this case.

https://github.com/ruby/ruby/blob/80045bc2a9e12012d6d16517ea5cb037e67eb2c1/test/ruby/test_range.rb#L333-L355

I don't know how to fix this issue.

But I think, This is a key sentence maybe.

Use alias.

```
rb_define_alias(rb_cRange, "===", "include?");
```

instead of

```
rb_define_method(rb_cRange, "===", range_include, 1);
```

#2 - 12/04/2016 01:49 AM - ksss (Yuki Kurihara)

Yes, I tried alias way.
But it was very slow.

```
$ time ./miniruby t.rb
./miniruby t.rb 1.80s user 0.02s system 98% cpu 1.842 total
```

And another issue appeared.

```
$ make test-all TESTS="test/ruby/test_range.rb"
CC = clang
LD = ld
LDSHARED = clang -dynamic -bundle
CFLAGS = -O0 -fno-fast-math -g3 -Wall -Wextra -Wno-unused-parameter -Wno-parentheses -Wno-long-long -Wno-missing-field-initializers -Wno-tautological-compare -Wno-parentheses-equality -Wno-constant-logical-operand -Wno-self-assign -Wno-unused-variable -Werror=implicit-int -Werror=pointer-arith -Werror=write-strings -Werror=declaration-after-statement -Werror=shorten-64-to-32 -Werror=implicit-function-declaration -Werror=division-by-zero -Werror=deprecated-declarations -Werror=extra-tokens -pipe
XCFLAGS = -fstack-protector -fno-strict-overflow -fvisibility=hidden -DRUBY_EXPORT -fPIE
CPPFLAGS = -D_XOPEN_SOURCE -D_DARWIN_C_SOURCE -D_DARWIN_UNLIMITED_SELECT -D_REENTRANT -I. -I.ext/include/x86_64-darwin15 -I./include -I. -I./enc/unicode/9.0.0
DLDFLAGS = -Wl,-undefined,dynamic_lookup -Wl,-multiply_defined,suppress -fstack-protector -Wl,-u,_objc_msgSend -Wl,-pie -framework CoreFoundation
SOLIBS = -lgmp
Apple LLVM version 8.0.0 (clang-800.0.42.1)
Target: x86_64-apple-darwin15.6.0
Thread model: posix
InstalledDir: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin
generating known_errors.inc
known_errors.inc unchanged
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common ./tool/runruby.rb --extout=.ext -- --disable-gems"
--excludes-dir=./test/excludes --name=!/memory_leak/
```

Running tests:

```
[21/42] TestRange#test_eqq_time = 0.00 s
1) Failure:
TestRange#test_eqq_time [/Users/yuki/src/github.com/ksss/ruby/test/ruby/test_range.rb:328]:
[ruby-core:69052] [Bug #11113].
Exception raised:
<#<TypeError: can't iterate from Time>>
```

#3 - 12/20/2016 03:34 AM - shyouhei (Shyouhei Urabe)

- Related to Feature #12612: Switch Range#=== to use cover? instead of include? added

#4 - 02/22/2017 05:54 AM - matz (Yukihiro Matsumoto)

Sounds reasonable. We'd like to see how big incompatibility is.

Matz.

#5 - 05/17/2017 02:14 PM - ksss (Yuki Kurihara)

- File range_eqq2.patch added

I updated the patch (range_eqq2.patch).

It optimize only special cases.

Range with Fixnum

```
i = 0
while i < 10_000_000
  case i
  when 1..1000
  end
  i += 1
end
```

Before

```
$ time ./miniruby t.rb
./miniruby t.rb 5.27s user 0.02s system 98% cpu 5.354 total
```

After

```
$ time ./miniruby t.rb
./miniruby t.rb 4.45s user 0.04s system 96% cpu 4.644 total
```

Range with String

```
i = 0
while i < 10_000_000
  case "c"
  when "a".."z"
  end
  i += 1
end
```

Before

```
$ time ./miniruby t.rb
./miniruby t.rb 8.45s user 0.02s system 99% cpu 8.534 total
```

After

```
$ time ./miniruby t.rb
./miniruby t.rb 3.80s user 0.01s system 98% cpu 3.874 total
```

Of course, test-all passed.

```
$ make test-all TESTS=test/ruby/test_range.rb
./tool/ifchange "--timestamp=rbconfig.time" rbconfig.rb rbconfig.tmp
rbconfig.rb unchanged
creating verconf.h
verconf.h updated
compiling loadpath.c
linking static-library libruby.2.5.0-static.a
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/ranlib: file: libruby.2
.5.0-static.a(debug_counter.o) has no symbols
linking shared-library libruby.2.5.0.dylib
linking ruby
Run options: "--ruby=./miniruby -I./lib -I. -I.ext/common ./tool/runruby.rb --extout=.ext -- --disable-gems"
--excludes-dir=./test/excludes --name=!/memory_leak/
```

```
# Running tests:
```

```
Finished tests in 2.777357s, 14.7622 tests/s, 144652.6320 assertions/s.
41 tests, 401752 assertions, 0 failures, 0 errors, 0 skips
```

```
ruby -v: ruby 2.5.0dev (2017-05-17 trunk 58600) [x86_64-darwin16]
```

#6 - 05/17/2018 07:08 AM - matz (Yukihiko Matsumoto)

- Related to Feature #14575: Switch Range#=== to use cover? instead of include? added

Files

range-eqq.patch	1.35 KB	12/02/2016	ksss (Yuki Kurihara)
range_eqq2.patch	1.2 KB	05/17/2017	ksss (Yuki Kurihara)