Ruby master - Bug #9676
String#gsub shouldn't allocate so many Strings in its loop
03/25/2014 10:55 PM - srawlins (Sam Rawlins)

Description
rb_reg_search() allocates (dups) a String to attach to the backreference object (RMATCH(match)->str = rb_str_new4(str);). If #gsub has been passed 2 arguments (not Enumerator form) and the second argument is a String, then it shouldn't make these allocations when calling rb_reg_search() inside it's loop.

Here's an example:

```ruby
# gsub-allocates-too-much.rb
require File.join(__dir__, "lib", "allocation_stats")

def puts_object_list(name, stats)
  objects = stats.allocations.group_by(:sourcefile, :sourceline, :class).all.
  values.flatten.map(&:object).
  map{|o| o.is_a?(String) ? "#{o.inspect}<#{o.encoding.to_s}>": o.inspect }
  puts "#{name} #{objects.flatten.size} new objects:
  objects.group_by(&:hash).values.each { |ary| puts "#{ary.join(", ")}" }
end

slash = '/'; underscore = '_'; colon = ':' # allocate before the trace
str = "12:34:45:67"
stats = AllocationStats.trace { str.gsub(colon, underscore) }
puts > "12:34:45:67".gsub(":\", "_")
puts_object_list("gsub substitutes 3x times:", stats)

$ ruby ../allocation_stats/gsub-allocates-too-much.rb
> "12:34:45:67".gsub(":\", "_")
gsub substitutes 3x times: 12 new objects:
  "1234567"<UTF-8>
  ":\"<ASCII-8BIT>, ":\"<ASCII-8BIT
  ":<US-ASCII>, ":<US-ASCII>
  #=>MatchData ":>
  #=>MatchData nil
  #:/

The Strings that are copies of the original String are all unnecessary (except one, the last).

I have a fix (attached and at [1]) that involves allocating the str attribute of the backreference object only when necessary. In order to do this without changing the signature of rb_reg_search(), this patch changes rb_reg_search() to wrap a new function rb_reg_search0(). So no calls to rb_reg_search() need to change, and str_gsub() changes two calls into rb_reg_search0() to avoid the allocations. (I believe String#split suffers from the same extra allocations, and can make a similar call to rb_reg_search0().)

The impact of this fix is primarily faster garbage collection. I have two "real world" examples:

- ActiveRecord sqlite3 specs: total time in GC reduced from 11.2s to 10.4s (7% savings).
- Mail gem specs: total time in GC reduced from 0.220s to 0.215s (2% savings).

These numbers bounced around a lot though. I'm open to better benchmarking suggestions. I used ActiveRecord and Mail for real world examples of #gsub, where realistic Strings are gsubbed.

[1] https://github.com/ruby/ruby/pull/578
Stop allocating backref strings within gsub's search loop

- internal.h: add prototype for rb_reg_search0

- re.c: rename rb_reg_search to rb_reg_search0, add set_backref_str argument to allow callers to indicate that they don't require the backref string to be allocated

- string.c: don't allocate backref str if replacement string is provided

Closes GH-578. [Bug #9676] [ruby-core:61682]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45414 b2dd03c8-39d4-4d8f-98ff-823fe69b080e
Closes GH-578. [Bug #9676] [ruby-core:61682]

Revision 45414 - 03/25/2014 11:46 PM - charliesome (Charlie Somerville)
Stop allocating backref strings within gsub's search loop

- internal.h: add prototype for rb_reg_search0
- re.c: rename rb_reg_search to rb_reg_search0, add set_backref_str argument to allow callers to indicate that they don't require the backref string to be allocated
- string.c: don't allocate backref str if replacement string is provided

Closes GH-578. [Bug #9676] [ruby-core:61682]

Revision 45414 - 03/25/2014 11:46 PM - charliesome (Charlie Somerville)
Stop allocating backref strings within gsub's search loop

- internal.h: add prototype for rb_reg_search0
- re.c: rename rb_reg_search to rb_reg_search0, add set_backref_str argument to allow callers to indicate that they don't require the backref string to be allocated
- string.c: don't allocate backref str if replacement string is provided

Closes GH-578. [Bug #9676] [ruby-core:61682]

History
#1 - 03/25/2014 11:46 PM - charliesome (Charlie Somerville)
- Status changed from Open to Closed
- % Done changed from 0 to 100

Applied in changeset r45414.

Stop allocating backref strings within gsub's search loop

- internal.h: add prototype for rb_reg_search0
- re.c: rename rb_reg_search to rb_reg_search0, add set_backref_str argument to allow callers to indicate that they don't require the backref string to be allocated
- string.c: don't allocate backref str if replacement string is provided

Closes GH-578. [Bug #9676] [ruby-core:61682]

#2 - 07/02/2014 06:40 AM - usa (Usaku NAKAMURA)
- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.0.0: WONTFIX, 2.1: UNKNOWN

Files
ruby-578.diff 2.56 KB 03/25/2014 srawlins (Sam Rawlins)