Ruby trunk - Feature #14326

[PATCH] net/protocol: read directly into rbuf if it's empty

01/07/2018 12:43 AM - normalperson (Eric Wong)

Status:	Closed
Priority:	Normal
Assignee:	naruse (Yui NARUSE)
Target version:	

Description

This relies on the reverted r61638 which was prematurely committed:
https://svn.ruby-lang.org/cgi-bin/viewvc.cgi?view=revision&revision=61638

I guess it was luck that @rbuf worked before with UTF-8 encoding.

```
net/protocol: read directly into rbuf if it's empty

There's no need to allocate a temporary string when @rbuf is
empty, we can use it as the read_nonblock destination buffer to
save both allocation overhead and avoid a later memcpy.

This results in a halving user CPU time and tiny memory
reduction with the script below:

         user     system      total        real
before   0.603333   0.539999   1.143332 (  1.143347)
       RssAnon:     5624 kB

after    0.283334   0.560000   0.843334 (  0.846072)
       RssAnon:     5592 kB

------
require 'net/http'
require 'benchmark'
s = TCPServer.new('127.0.0.1', 0)
len = 1024 * 1024 * 1024 * 2
pid = fork do
  c = s.accept
  c.readpartial(16384).clear
  c.send("HTTP/1.0 200 OK\r\nContent-Length: #{len}\r\n\r\n", Socket::MSG_MORE)
  IO.copy_stream('/dev/zero', c, len)
  c.close
end

addr = s.addr
Net::HTTP.start(addr[3], addr[1]) do |http|
  http.request_get('/') do |res|
    puts(Benchmark.measure { res.read_body(&:clear) })
  end
end
puts File.readlines("/proc/self/status").grep(/RssAnon/)[0]
Process.waitpid2(pid)
------

* lib/net/protocol.rb (rbuf_fill): avoid allocation if rbuf is empty
```

Associated revisions

Revision b02fc0f9 - 01/08/2018 12:34 AM - normal

net/protocol: read directly into rbuf if it's empty

There's no need to allocate a temporary string when @rbuf is
empty, we can use it as the read_nonblock destination buffer to
save both allocation overhead and avoid a later memcpy.

This results in a halving user CPU time and tiny memory
reduction with the script below:

```
          user     system      total        real
```

before   0.603333   0.539999   1.143332 (1.143347)
RssAnon: 5624 kB

after   0.283334   0.560000   0.843334 (0.846072)
RssAnon: 5592 kB

```
require 'net/http'
require 'benchmark'
s = TCPServer.new('127.0.0.1', 0)
len = 1024 * 1024 * 1024 * 2
pid = fork do
c = s.accept
c.readpartial(16384).clear
c.send("HTTP/1.0 200 OK\r\nContent-Length: #{len}\r\n\r\n", Socket::MSG_MORE)
IO.copy_stream('/dev/zero', c, len)
c.close
end

addr = s.addr
Net::HTTP.start(addr[3], addr[1]) do |http|
http.request_get('/') do |res|
puts(Benchmark.measure { res.read_body(&:clear) })
end
end
puts File.readlines("/proc/self/status").grep(/RssAnon/)[0]
```

Process.waitpid2(pid)

- lib/net/protocol.rb (rbuf_fill): avoid allocation if rbuf is empty [ruby-core:84678] [Feature #14326]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@61663 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 61663 - 01/08/2018 12:34 AM - normalperson (Eric Wong)

net/protocol: read directly into rbuf if it's empty

There's no need to allocate a temporary string when @rbuf is
empty, we can use it as the read_nonblock destination buffer to
save both allocation overhead and avoid a later memcpy.

This results in a halving user CPU time and tiny memory
reduction with the script below:

```
          user     system      total        real
```

before   0.603333   0.539999   1.143332 (1.143347)
RssAnon: 5624 kB

after   0.283334   0.560000   0.843334 (0.846072)
RssAnon: 5592 kB

```
require 'net/http'
require 'benchmark'
s = TCPServer.new('127.0.0.1', 0)
len = 1024 * 1024 * 1024 * 2
pid = fork do
c = s.accept
c.readpartial(16384).clear
c.send("HTTP/1.0 200 OK\r\nContent-Length: #{len}\r\n\r\n", Socket::MSG_MORE)
IO.copy_stream('/dev/zero', c, len)
c.close
end

addr = s.addr
Net::HTTP.start(addr[3], addr[1]) do |http|
http.request_get('/') do |res|
puts(Benchmark.measure { res.read_body(&:clear) })
end
```

```
end
puts File.readlines("/proc/self/status").grep(/RssAnon/)[0]
```

Process.waitpid2(pid)

- lib/net/protocol.rb (rbuf_fill): avoid allocation if rbuf is empty [ruby-core:84678] [Feature #14326]

Revision 61663 - 01/08/2018 12:34 AM - normal

net/protocol: read directly into rbuf if it's empty

There's no need to allocate a temporary string when @rbuf is
empty, we can use it as the read_nonblock destination buffer to
save both allocation overhead and avoid a later memcpy.

This results in a halving user CPU time and tiny memory
reduction with the script below:

```
              user     system      total        real
```

```
before   0.603333   0.539999   1.143332 (  1.143347)
RssAnon:     5624 kB

after    0.283334   0.560000   0.843334 (  0.846072)
RssAnon:     5592 kB
```

```
require 'net/http'
require 'benchmark'
s = TCPServer.new('127.0.0.1', 0)
len = 1024 * 1024 * 1024 * 2
pid = fork do
c = s.accept
c.readpartial(16384).clear
c.send("HTTP/1.0 200 OK\r\nContent-Length: #{len}\r\n\r\n", Socket::MSG_MORE)
IO.copy_stream('/dev/zero', c, len)
c.close
end

addr = s.addr
Net::HTTP.start(addr[3], addr[1]) do |http|
http.request_get('/') do |res|
puts(Benchmark.measure { res.read_body(&:clear) })
end
end
puts File.readlines("/proc/self/status").grep(/RssAnon/)[0]
```

Process.waitpid2(pid)

- lib/net/protocol.rb (rbuf_fill): avoid allocation if rbuf is empty [ruby-core:84678] [Feature #14326]

History

#1 - 01/08/2018 12:34 AM - Anonymous

- Status changed from Open to Closed

Applied in changeset trunk|r61663.

---

net/protocol: read directly into rbuf if it's empty

There's no need to allocate a temporary string when @rbuf is
empty, we can use it as the read_nonblock destination buffer to
save both allocation overhead and avoid a later memcpy.

This results in a halving user CPU time and tiny memory
reduction with the script below:

```
              user     system      total        real
```

```
before   0.603333   0.539999   1.143332 (  1.143347)
RssAnon:     5624 kB
```

```
after    0.283334   0.560000   0.843334 (  0.846072)
RssAnon:     5592 kB
```

```
require 'net/http'
require 'benchmark'
s = TCPServer.new('127.0.0.1', 0)
len = 1024 * 1024 * 1024 * 2
pid = fork do
c = s.accept
c.readpartial(16384).clear
c.send("HTTP/1.0 200 OK\r\nContent-Length: #{len}\r\n\r\n", Socket::MSG_MORE)
IO.copy_stream('/dev/zero', c, len)
c.close
end

addr = s.addr
Net::HTTP.start(addr[3], addr[1]) do |http|
http.request_get('/') do |res|
puts(Benchmark.measure { res.read_body(&:clear) })
end
end
puts File.readlines("/proc/self/status").grep(/RssAnon/)[0]
```

Process.waitpid2(pid)

- lib/net/protocol.rb (rbuf_fill): avoid allocation if rbuf is empty [ruby-core:84678] [Feature #14326]