Ruby master - Bug #14908

Enumerator::Lazy creates unnecessary Array objects.

07/11/2018 01:23 PM - chopraanmol1 (Anmol Chopra)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 2.6.0dev (2018-07-11 trunk 63949) [x86_64-linux]</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN</td>
</tr>
</tbody>
</table>

Description

Benchmark result on trunk:

<table>
<thead>
<tr>
<th>user</th>
<th>system</th>
<th>total</th>
<th>real</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lazy:</td>
<td>0.120000</td>
<td>0.000000</td>
<td>0.120000 (0.119958)</td>
</tr>
<tr>
<td>Normal:</td>
<td>0.056000</td>
<td>0.004000</td>
<td>0.060000 (0.062848)</td>
</tr>
<tr>
<td></td>
<td>2.142857</td>
<td>0.000000</td>
<td>NaN (1.908698)</td>
</tr>
</tbody>
</table>

Lazy:
Total allocated: 122240 bytes (3033 objects)
Total retained: 0 bytes (0 objects)

allocated memory by class

+----------------------------------+-
| 120480 | Array |
| 880    | Proc  |
| 384    | Enumerator::Lazy |
| 264    | Object |
| 168    | Enumerator::Generator |
| 64     | Enumerator::Yielder |

allocated objects by class

+----------------------------------+-
| 3012 | Array |
| 11   | Proc  |
| 3    | Enumerator::Generator |
| 3    | Enumerator::Lazy |
| 3    | Object |
| 1    | Enumerator::Yielder |

Normal:
Total allocated: 72120 bytes (3 objects)
Total retained: 0 bytes (0 objects)

allocated memory by class

+----------------------------------+-
| 72120 | Array |

allocated objects by class

+----------------------------------+-
| 3 | Array |

As you may observe an extra array is created for every final element. Enumerator::Yielder#yield method has arity of -1 which wrap every elements in array. The same goes for Enumerator::Yielder#<< method, I'm proposing to change arity for Enumerator::Yielder#<< to 1 from -1 and use it internally for the lazy enum. It will also make << method definition consistent with other classes(Array, String & etc).

I've applied the following set of changes:

[https://github.com/ruby/ruby/pull/1912](https://github.com/ruby/ruby/pull/1912)
which result in the following:

<table>
<thead>
<tr>
<th></th>
<th>user</th>
<th>system</th>
<th>total</th>
<th>real</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lazy:</td>
<td>0.108000</td>
<td>0.000000</td>
<td>0.108000</td>
<td>0.108484</td>
</tr>
<tr>
<td>Normal:</td>
<td>0.052000</td>
<td>0.008000</td>
<td>0.060000</td>
<td>0.062528</td>
</tr>
</tbody>
</table>

2.076923 0.000000 NaN ( 1.734961)

Lazy:
Total allocated: 2240 bytes (33 objects)
Total retained: 0 bytes (0 objects)

allocated memory by class
--------------------------------------------------
880 Proc
480 Array
384 Enumerator::Lazy
264 Object
168 Enumerator::Generator
64 Enumerator::Yielder

allocated objects by class
--------------------------------------------------
12 Array
11 Proc
3 Enumerator::Generator
3 Enumerator::Lazy
3 Object
1 Enumerator::Yielder

Normal:
Total allocated: 72120 bytes (3 objects)
Total retained: 0 bytes (0 objects)

allocated memory by class
--------------------------------------------------
72120 Array

allocated objects by class
--------------------------------------------------
3 Array

This changes reduces the memory utilization and also by a tiny fraction improves performance for the lazy enumerator

Associated revisions
Revision aefdd4e2 - 09/18/2018 08:49 AM - nobu (Nobuyoshi Nakada)
Lazy Enumerator reduce intermediate array creation
[ruby-core:87907] [Bug #14908] [Fix GH-1912]
From: Anmol Chopra chopraanmol1@gmail.com
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64770 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 64770 - 09/18/2018 08:49 AM - nobu (Nobuyoshi Nakada)
Lazy Enumerator reduce intermediate array creation
[ruby-core:87907] [Bug #14908] [Fix GH-1912]
From: Anmol Chopra chopraanmol1@gmail.com

Revision 64770 - 09/18/2018 08:49 AM - nobu (Nobuyoshi Nakada)
Lazy Enumerator reduce intermediate array creation
[ruby-core:87907] [Bug #14908] [Fix GH-1912]
#1 - 07/11/2018 02:23 PM - chopraanmol1 (Anmol Chopra)
- Description updated

#2 - 07/11/2018 02:29 PM - chopraanmol1 (Anmol Chopra)
- Description updated

#3 - 07/12/2018 12:12 AM - shyouhei (Shyouhei Urabe)
Understand the problem. The proposed fix however involves spec change. We need to discuss effects of it before applying this patch.

#4 - 07/12/2018 05:03 AM - chopraanmol1 (Anmol Chopra)
shyouhei (Shyouhei Urabe) wrote:

Understand the problem. The proposed fix however involves spec change. We need to discuss effects of it before applying this patch.

If this is a big breaking change than alternate will be to create a different method for Enumerator::Lazy's internal use. I'm also up for updating the patch to reflect that. But I think for future release it will make more sense to have Enumerator::Yielder#<< to have an arity of 1 if you consider syntax use case.

But for the backporting purpose, I'm more inclined to create a new method.

#5 - 07/12/2018 09:25 AM - Eregon (Benoit Daloze)
Changing Enumerator::Yielder#<< to have arity 1 seems fine to me, as I guess nobody calls << on an Enumerator::Yielder with more than 1 argument, isn't it?

#6 - 07/13/2018 04:46 AM - chopraanmol1 (Anmol Chopra)
Eregon (Benoit Daloze) wrote:

Changing Enumerator::Yielder#<< to have arity 1 seems fine to me, as I guess nobody calls << on an Enumerator::Yielder with more than 1 argument, isn't it?

Yes, that will be the general case. Exception:

```
.send({<<,...})
<<(...)
```

#7 - 07/13/2018 04:59 AM - marcandre (Marc-Andre Lafortune)
Indeed, as long as Yielder#yield is kept with arity -1 (as in this patch), indeed I don't think that would be an "incompatibility" we should worry about.

#8 - 07/13/2018 07:28 AM - chopraanmol1 (Anmol Chopra)
- Description updated

#9 - 08/16/2018 06:17 AM - chopraanmol1 (Anmol Chopra)
- Description updated

#10 - 09/13/2018 07:53 AM - knu (Akinori MUSHA)
I'm the original author of Enumerator & Yielder and I don't think I particularly intended to make << accept many arguments. << was an alias of yield, and I just didn't bother to make a separate function. So, I'd say go ahead.

#11 - 09/18/2018 05:48 AM - chopraanmol1 (Anmol Chopra)
All the responses for the proposed patch are positive till date. @shyouhei (Shyouhei Urabe) can we go ahead with this now?

#12 - 09/18/2018 06:43 AM - shyouhei (Shyouhei Urabe)
No objection. I guess @nobu (Nobuyoshi Nakada) is trying to fix it.

#13 - 09/18/2018 08:49 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed
Lazy Enumerator reduce intermediate array creation

From: Anmol Chopra chopraanmol1@gmail.com

Files

| lazy_test.rb | 1.26 KB | 07/11/2018 | chopraanmol1 (Anmol Chopra) |