

Ruby master - Feature #13342

Improve yielding block performance

03/21/2017 12:47 AM - watson1978 (Shizuo Fujita)

Status:	Open																				
Priority:	Normal																				
Assignee:																					
Target version:																					
Description																					
The yielding block will be faster around 9%. This patch ensures that expand to inline codes in where invoke yielding block.																					
Environment																					
<ul style="list-style-type: none">• macOS 10.12.3• clang 8.0.0 in Xcode 8.2																					
Before																					
<table><thead><tr><th></th><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>Integer#times</td><td>0.930000</td><td>0.000000</td><td>0.930000 (</td><td>0.932125)</td></tr><tr><td>Array#each</td><td>0.950000</td><td>0.000000</td><td>0.950000 (</td><td>0.957962)</td></tr><tr><td>Array#map</td><td>1.220000</td><td>0.030000</td><td>1.250000 (</td><td>1.249174)</td></tr></tbody></table>			user	system	total	real	Integer#times	0.930000	0.000000	0.930000 (0.932125)	Array#each	0.950000	0.000000	0.950000 (0.957962)	Array#map	1.220000	0.030000	1.250000 (1.249174)
	user	system	total	real																	
Integer#times	0.930000	0.000000	0.930000 (0.932125)																	
Array#each	0.950000	0.000000	0.950000 (0.957962)																	
Array#map	1.220000	0.030000	1.250000 (1.249174)																	
After																					
<table><thead><tr><th></th><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>Integer#times</td><td>0.850000</td><td>0.000000</td><td>0.850000 (</td><td>0.853202)</td></tr><tr><td>Array#each</td><td>0.860000</td><td>0.010000</td><td>0.870000 (</td><td>0.865507)</td></tr><tr><td>Array#map</td><td>1.120000</td><td>0.020000</td><td>1.140000 (</td><td>1.149939)</td></tr></tbody></table>			user	system	total	real	Integer#times	0.850000	0.000000	0.850000 (0.853202)	Array#each	0.860000	0.010000	0.870000 (0.865507)	Array#map	1.120000	0.020000	1.140000 (1.149939)
	user	system	total	real																	
Integer#times	0.850000	0.000000	0.850000 (0.853202)																	
Array#each	0.860000	0.010000	0.870000 (0.865507)																	
Array#map	1.120000	0.020000	1.140000 (1.149939)																	
Test code																					
<pre>require 'benchmark' Benchmark.bmbm do x ary = (1..10000).to_a x.report "Integer#times" do 20000000.times do end end x.report "Array#each" do 2000.times do ary.each { x } end end x.report "Array#map" do 2000.times do ary.map { x } end end end</pre>																					
Patch																					
The patch is in https://github.com/ruby/ruby/pull/1535																					

History

#1 - 04/14/2017 08:40 AM - watson1978 (Shizuo Fujita)

When I filed this ticket, I tried to run benchmark on macOS + clang only.
Then, I tried to do on 2 environments in additional.

macOS 10.12 + gcc 6.3.0

Before

	user	system	total	real
Integer#times	0.890000	0.000000	0.890000 (0.886378)
Array#each	0.840000	0.010000	0.850000 (0.849755)
Array#map	1.070000	0.030000	1.100000 (1.094257)

After

	user	system	total	real
Integer#times	0.860000	0.000000	0.860000 (0.860164)
Array#each	0.870000	0.000000	0.870000 (0.867758)
Array#map	1.070000	0.030000	1.100000 (1.102324)

Ubuntu 16.04.4 + gcc 5.4.0

Before

	user	system	total	real
Integer#times	0.560000	0.000000	0.560000 (0.561627)
Array#each	0.550000	0.000000	0.550000 (0.552468)
Array#map	0.710000	0.000000	0.710000 (0.714104)

After

	user	system	total	real
Integer#times	0.530000	0.000000	0.530000 (0.533428)
Array#each	0.530000	0.000000	0.530000 (0.526173)
Array#map	0.650000	0.020000	0.670000 (0.676441)

#2 - 06/24/2019 08:33 PM - jeremyevans0 (Jeremy Evans)

- Backport deleted (2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN)

- Tracker changed from Bug to Feature