

Ruby master - Feature #13482

Improve performance of "set instance variable"

04/19/2017 04:31 AM - watson1978 (Shizuo Fujita)

Status: Open									
Priority: Normal									
Assignee:									
Target version:									
Description									
"set instance variable" will be faster around 14%.									
macOS 10.12 + clang-802.0.41									
Before									
<table><thead><tr><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>1.550000</td><td>0.000000</td><td>1.550000</td><td>(1.555161)</td></tr></tbody></table>		user	system	total	real	1.550000	0.000000	1.550000	(1.555161)
user	system	total	real						
1.550000	0.000000	1.550000	(1.555161)						
After									
<table><thead><tr><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>1.400000</td><td>0.000000</td><td>1.400000</td><td>(1.398825)</td></tr></tbody></table>		user	system	total	real	1.400000	0.000000	1.400000	(1.398825)
user	system	total	real						
1.400000	0.000000	1.400000	(1.398825)						
macOS 10.12 + gcc 6.3.0_1									
Before									
<table><thead><tr><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>1.660000</td><td>0.000000</td><td>1.660000</td><td>(1.659721)</td></tr></tbody></table>		user	system	total	real	1.660000	0.000000	1.660000	(1.659721)
user	system	total	real						
1.660000	0.000000	1.660000	(1.659721)						
After									
<table><thead><tr><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>1.450000</td><td>0.000000</td><td>1.450000</td><td>(1.454589)</td></tr></tbody></table>		user	system	total	real	1.450000	0.000000	1.450000	(1.454589)
user	system	total	real						
1.450000	0.000000	1.450000	(1.454589)						
Ubuntu 16.04 + gcc 5.4.0									
Before									
<table><thead><tr><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>0.890000</td><td>0.000000</td><td>0.890000</td><td>(0.891217)</td></tr></tbody></table>		user	system	total	real	0.890000	0.000000	0.890000	(0.891217)
user	system	total	real						
0.890000	0.000000	0.890000	(0.891217)						
After									
<table><thead><tr><th>user</th><th>system</th><th>total</th><th>real</th></tr></thead><tbody><tr><td>0.790000</td><td>0.000000</td><td>0.790000</td><td>(0.783157)</td></tr></tbody></table>		user	system	total	real	0.790000	0.000000	0.790000	(0.783157)
user	system	total	real						
0.790000	0.000000	0.790000	(0.783157)						
Test code									
<pre>require 'benchmark' Benchmark.bmbm do x x.report do i = 0 while (i < 50000000) @obj = 42 i+=1 end end end</pre>									

end

Patch

<https://github.com/ruby/ruby/pull/1590>

History

#1 - 04/19/2017 04:32 AM - watson1978 (Shizuo Fujita)

- *Description updated*

#2 - 04/19/2017 09:35 AM - watson1978 (Shizuo Fujita)

It has already ensured inlining in where `vm_getivar()`

(https://github.com/ruby/ruby/blob/524fb0138b773f2ed01441abbcffeda0271175c5/vm_inshelper.c#L908)

So, I'm guessing that there is no reason what avoid inlining at `vm_setivar()`

#3 - 06/24/2019 08:25 PM - jeremyevans0 (Jeremy Evans)

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

- *Tracker changed from Bug to Feature*