

Ruby - Bug #12074

[PERF] bm_app_erb.rb slower

02/16/2016 02:04 AM - tgxworld (Guo Xiang Tan)

Status:	Closed	Backport: 2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN
Priority:	Normal	
Assignee:	k0kubun (Takashi Kokubun)	
Target version:		
ruby -v:	ruby 2.4.0dev (2016-01-02 trunk 53412)	
Description		
Hi I'm noticing a regression on RubyBench for bm_app_erb.rb		
http://rubybench.org/ruby/ruby/commits?result_type=app_erb&display_count=2000		
Narrowed down to https://github.com/ruby/ruby/commit/32b4a8b		
Results on my local machine confirms it as well:		
<pre>total: 2 trial(s) (2 trial(s) for 1 benchmark(s)) .. ----- benchmark results: Execution time (sec) name ruby 2.4.0dev (2016-01-02 trunk 53411) [x86_64-linux] Posting results to Web UI.... app_erb 0.614 total: 2 trial(s) (2 trial(s) for 1 benchmark(s)) .. ----- benchmark results: Execution time (sec) name ruby 2.4.0dev (2016-01-02 trunk 53412) [x86_64-linux] Posting results to Web UI.... app_erb 0.911</pre>		
Let me know if I can provide more information :)		

Associated revisions

Revision 72fb1e1690d71f9ec9e33bc3bc282689c7afe2b4 - 05/22/2017 03:56 AM - k0kubun (Takashi Kokubun)

erb.rb: Skip creating regexp

if stags and etags are not changed from default.

:putobject insn (of regexp) will be used instead of :toregexp insn.

This means that the regexp won't be compiled for every SimpleScanner#scan call.

It may not be a good idea to apply this kind of optimization for all cases.

But I applied this because it is default scanner and used frequently and has relatively large impact for benchmark like this:

- Before
app_erb 1.023
- After
app_erb 0.781

This commit fixes only the bottleneck of performance regression introduced in r53412. For maintainability, I won't fix other small regressions like additional overhead of method calls.

[ruby-core:73820] [Bug #12074]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@58842 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 72fb1e16 - 05/22/2017 03:56 AM - k0kubun (Takashi Kokubun)

erb.rb: Skip creating regexp

if stags and etags are not changed from default.

:putobject insn (of regexp) will be used instead of :toregexp insn.
This means that the regexp won't be compiled for every
SimpleScanner#scan call.

It may not be a good idea to apply this kind of optimization for all cases.
But I applied this because it is default scanner and used frequently and has
relatively large impact for benchmark like this:

- Before
app_erb 1.023
- After
app_erb 0.781

This commit fixes only the bottleneck of performance regression introduced
in r53412. For maintainability, I won't fix other small regressions like
additional overhead of method calls.

[ruby-core:73820] [Bug #12074]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@58842 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 05/20/2017 04:24 PM - k0kubun (Takashi Kokubun)

- Assignee set to k0kubun (Takashi Kokubun)

#2 - 05/22/2017 03:56 AM - k0kubun (Takashi Kokubun)

- Status changed from Open to Closed

Applied in changeset trunk|r58842.

erb.rb: Skip creating regexp

if stags and etags are not changed from default.

:putobject insn (of regexp) will be used instead of :toregexp insn.
This means that the regexp won't be compiled for every
SimpleScanner#scan call.

It may not be a good idea to apply this kind of optimization for all cases.
But I applied this because it is default scanner and used frequently and has
relatively large impact for benchmark like this:

- Before
app_erb 1.023
- After
app_erb 0.781

This commit fixes only the bottleneck of performance regression introduced
in r53412. For maintainability, I won't fix other small regressions like
additional overhead of method calls.

[\[ruby-core:73820\]](#) [\[Bug #12074\]](#)