

Ruby trunk - Bug #6250

Enumerator::Lazy performance increased

04/03/2012 05:52 PM - gregolsen (Innokenty Mikhailov)

Status: Rejected	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.0.0dev (2012-04-03 trunk 35220) [x86_64-linux]	Backport:
Description =begin I'm terribly sorry but it seems that I can't reopen existing issue (https://bugs.ruby-lang.org/issues/6183), so here's the new one: Finally come up with a concrete idea how to "fix" lazy enumerator performance (based on my first PR https://github.com/ruby/ruby/pull/100). The idea is to keep all blocks (passed with lazy methods like map or select) as Proc objects inside the enumerator and apply them one by one when value requested (to_a, next, etc) This strategy avoids enumerator chaining on each lazy method call and eliminates fair amount of 'calling the block' with rb_block_call operations. Here's benchmark results: 2.0.0 ~/projects/ruby(trunk)\$ rvm ruby-head 2.0.0 ~/projects/ruby(trunk)\$ ruby bench.rb user system total real Lazy enumerator 1.460000 0.000000 1.460000 (1.465739) Simple array 0.420000 0.000000 0.420000 (0.421446) 0.287671 NaN NaN (0.287531) 2.0.0 ~/projects/ruby(trunk)\$ rvm system 2.0.0 ~/projects/ruby(trunk)\$ ruby bench.rb user system total real Lazy enumerator 0.770000 0.000000 0.770000 (0.764750) Simple array 0.370000 0.000000 0.370000 (0.382653) 0.480519 NaN NaN (0.500364) ruby-head is current trunk compiled, and system ruby - is the same trunk but with my patch applied. Last row in results is ratio between 'Simple array' time and 'Lazy Enumerator' time. So, as you can see, with this patch lazy enumerator becomes almost 2 times faster. It's a 'proof of concept' patch (only map and select added) - let me know if it makes sense. I believe that using this approach and with your help lazy enumerator performance can be improved significantly. I'm attaching the diff along with the main part of the source code just in case it's hard to follow the diff. Thanks. =end	
Related issues: Is duplicate of Ruby trunk - Feature #6183: Enumerator::Lazy performance issue Closed	

History

#1 - 04/03/2012 06:36 PM - mame (Yusuke Endoh)

- Status changed from Open to Rejected

[#6183](#) is reopened now. Please discuss it in that ticket. Thanks,

--

Yusuke Endoh mame@tsg.ne.jp

#2 - 04/03/2012 10:22 PM - trans (Thomas Sawyer)

In response to [#6250](#), wouldn't the simplest implementation be to return a new Lazy Enumerator for each lazy call?

See <https://github.com/rubyworks/facets/blob/master/lib/core/facets/denumerable.rb> for the basic idea.

Maybe storing each block internally in an Enumerator might be a tad more efficient, but I would think the added complexity to Enumerator state would make it not worth it.

Files

lazy_enumerator.diff	28.6 KB	04/03/2012	gregolsen (Innokenty Mikhailov)
code	2.78 KB	04/03/2012	gregolsen (Innokenty Mikhailov)
bench.rb	389 Bytes	04/03/2012	gregolsen (Innokenty Mikhailov)