Ruby trunk - Feature #3322

Simple Patch to make ruby copy-on-write-friendly

05/20/2010 05:58 PM - Dan42 (Daniel DeLorme)

Status:	Rejected
Priority:	Normal
Assignee:	authorNari (Narihiro Nakamura)
Target version:	

Description

=begin
A lot has been said about ruby's lack of COW-friendliness; a full solution like REE has a performance impact, but I think I've come up with a simple trick that solves the problem of pre-forking servers. Basically, this patch implements GC.leak which marks all existing objects as ineligible for GC, and existing heaps as ineligible for allocation of new objects. It may sound hacky, but it's exactly the point of pre-forking in most cases: load a bunch of code & data and then keep it in memory forever.

Compare the memory usage of these:
./ruby -e 'x=(1..40_000_000).map{""};   Process.fork; x=[]; GC.start; sleep 10'
./ruby -e 'x=(1..40_000_000).map{""}; GC.leak; Process.fork; x=[]; GC.start; sleep 10'

This patch has the advantage of being
1) simple
2) free of negative performance impact
3) implemented and available now

PS: As far as I can tell the patch works fine, but I'm not terribly well-versed about the innards of ruby so I'd appreciate some code review. Thanks.
=end

History

#1 - 05/20/2010 06:11 PM - rkh (Konstantin Haase)

=begin
You might want to run the GC first.
=end

#2 - 05/21/2010 08:55 AM - Dan42 (Daniel DeLorme)

- File cow2.patch added

=begin
Adding garbage_collect() before leak and after unleak
=end

#3 - 05/22/2010 02:19 AM - rogerdpack (Roger Pack)

=begin
Good idea. Unfortunately it needs to traverse (and mark) all the "old" heaps during GC, so that it can see which items are still accessible in the namespace. Unless you can guarantee that no new objects will be pointed to by members of the old heap, you still have to traverse everything. At least AFAIK.
-rp
=end

#4 - 06/26/2011 01:46 PM - akr (Akira Tanaka)

- Project changed from Ruby to Ruby trunk

- Category changed from core to core

#5 - 03/18/2012 03:50 PM - nahi (Hiroshi Nakamura)

- Description updated

- Assignee set to authorNari (Narihiro Nakamura)

#6 - 03/18/2012 06:46 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#7 - 05/28/2012 08:32 PM - authorNari (Narihiro Nakamura)

- Status changed from Assigned to Rejected

- % Done changed from 0 to 100

Sorry for the late relpy.
We already have Bitmap Marking GC in Ruby2.0 so I reject it.

Thanks.

Files

cow.patch	2.41 KB	05/20/2010	Dan42 (Daniel DeLorme)
cow2.patch	2.46 KB	05/21/2010	Dan42 (Daniel DeLorme)