DRb.stop_service doesn't kill sleeping TimerIdConv threads

05/03/2016 03:17 PM - jrafanie (Joe Rafaniello)

Status: Closed
Priority: Normal
Assignee: Joe Rafaniello
Target version: ruby -v:
  ruby 2.3.1p112 (2016-04-26 revision 54768) [x86_64-darwin15]
Backport: 2.1: WONTFIX, 2.2: DONE, 2.3: DONE

Description

According to this commit[1], it's a good idea for DRb.stop_service to kill threads it creates.

My problem is I have a workflow that creates a new DRb server and provides it a TimerIdConv for keeping drb objects alive until it's not accessed for X seconds. It then stops the service.

This workflow can happen many times.

The TimerIdConv.new(5) call creates a new thread each time that never exits.

There isn't a public interface to kill these threads so I have to use instance variables to get the thread so it can be killed.

Also, you can't reuse the TimerIdConv because you don't know how long it has been sleeping and there is not a way to reset it to ensure it sleeps the full timeout.

I would prefer if I could tell DRb.start_service to use a :timer_idconv => 5, have it create the TimerIdConv and be responsible for killing it when I call stop_service.

See code below. I have tested with ruby 2.3.1 and the code below slowly continues to grow in thread count.

[1] Commit: c474ecb0dfe0dd0d5e6b2b41f09eaf251d0c7079 or https://github.com/ruby/ruby/commit/c474ecb0dfe0dd0d5e6b2b41f09eaf251d0c7079#diff-140ee364bf64098bb12e0a96fadb402

```ruby
require 'drb'
require 'drb/timeridconv'

100.times do
  # setup the drb server with a one second timer
  # creates a new thread that never dies
  timer = DRb::TimerIdConv.new(5)

  service = DRb.start_service("druby://127.0.0.1:0", nil, :idconv => timer)

  # ... some code that uses the DRb service

  # stop the drb server now that we're done with it
  service.stop_service

  # Below is ugly code that kills the timer thread. Without the code below, the
  # thread stays alive forever. This should have a public API.
  # It would be nice if DRb.start_service
  # accepted a :timer_idconv => 5 option that would create the timer thread itself
  # and was responsible for killing this thread when calling stop_service.

  # holder = timer.instance_variable_get(:@holder)
  # keeper = holder.instance_variable_get(:@keeper) if holder
  # keeper.kill if keeper
  puts Thread.list.length
  sleep 2
end
```

Associated revisions

03/09/2022
Revision e143a741 - 05/15/2016 11:59 AM - seki (Masatoshi Seki)
don't use keeper thread. [Bug #12342]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@55008 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 55008 - 05/15/2016 11:59 AM - seki (Masatoshi Seki)
don't use keeper thread. [Bug #12342]

Revision 55008 - 05/15/2016 11:59 AM - seki (Masatoshi Seki)
don't use keeper thread. [Bug #12342]

Revision 55008 - 05/15/2016 11:59 AM - seki (Masatoshi Seki)
don't use keeper thread. [Bug #12342]

Revision 55008 - 05/15/2016 11:59 AM - seki (Masatoshi Seki)
don't use keeper thread. [Bug #12342]

Revision c20b07d5 - 08/11/2016 06:30 PM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 55008: [Backport #12342]
  * lib/drb/timeridconv.rb: don't use keeper thread. [Bug #12342]
  * test/drb/ut_timerholder.rb: ditto.
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_3@55867 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 55867 - 08/11/2016 06:30 PM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 55008: [Backport #12342]
  * lib/drb/timeridconv.rb: don't use keeper thread. [Bug #12342]
  * test/drb/ut_timerholder.rb: ditto.

Revision 0d09e1af - 08/16/2016 05:53 AM - usa (Usaku NAKAMURA)
merge revision(s) 55008: [Backport #12342]
  * lib/drb/timeridconv.rb: don't use keeper thread. [Bug #12342]
  * test/drb/ut_timerholder.rb: ditto.
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_2@55929 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 55929 - 08/16/2016 05:53 AM - usa (Usaku NAKAMURA)
merge revision(s) 55008: [Backport #12342]
  * lib/drb/timeridconv.rb: don't use keeper thread. [Bug #12342]
  * test/drb/ut_timerholder.rb: ditto.

History
#1 - 05/03/2016 03:20 PM - jrafanie (Joe Rafaniello)
Here is a link to our workaround: https://github.com/ManageIQ/manageiq/pull/8399

#2 - 05/06/2016 02:08 PM - jrafanie (Joe Rafaniello)
I have opened a pull request to expose an API for killing the timer thread. This PR should be enough to workaround this issue for me, although it would be nice if DRb.start_service could be responsible for creating/starting the timer and stop_service could then automatically stop any timer thread created by the service:

https://github.com/ruby/ruby/pull/1342

#3 - 05/15/2016 11:59 AM - seki (Masatoshi Seki)
- Status changed from Open to Closed
don't use keeper thread. [Bug #12342]

#4 - 06/08/2016 05:04 AM - usa (Usaku NAKAMURA)
- Backport changed from 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN to 2.1: WONTFIX, 2.2: REQUIRED, 2.3: REQUIRED

#5 - 08/11/2016 06:30 PM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.1: WONTFIX, 2.2: REQUIRED, 2.3: REQUIRED to 2.1: WONTFIX, 2.2: REQUIRED, 2.3: DONE

ruby_2_3 r55867 merged revision(s) 55008.

#6 - 08/16/2016 05:53 AM - usa (Usaku NAKAMURA)
- Backport changed from 2.1: WONTFIX, 2.2: REQUIRED, 2.3: DONE to 2.1: WONTFIX, 2.2: DONE, 2.3: DONE

ruby_2_2 r55929 merged revision(s) 55008.