Ruby trunk - Bug #15711

Remove use of _id2ref from DRb

03/19/2019 03:56 PM - headius (Charles Nutter)

Status:	Closed		
Priority:	Normal		
Assignee:	seki (Masatoshi Seki)		
Target version:			
ruby -v:		**Backport:**	2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

Description

This issue relates to https://bugs.ruby-lang.org/issues/15408

DRb uses _idref internally to implement a weak map, and this issue seeks to replace that code with an implementation that does not use _id2ref.

We will be deprecating ObjectSpace._id2ref in the near future since it fails to work like people expect (when implemented as a pointer address) or adds memory and invocation overhead to object_id.

An initial patch for this is provided by JRuby, which implements object_id using a monotonically-increasing value, and only allows _id2ref use with a command line flag.

https://github.com/ruby/ruby/compare/trunk...jruby:jruby-ruby_2_6_0#diff-e979bf2f831d9826629559b8628809e9

This implementation uses the stdlib weakref to implement a simple weak map, and it would be suitable as an implementation for now. However there's some inefficiency here because it has to periodically "clean" the hash of vacated references by scanning all entries.

There are two more efficient implementations that require additional work:

Alternate 1: Use ObjectSpace::WeakMap, which is an opaque VM-supported implementation of a weak Hash. Unfortunately I don't think WeakMap has ever been blessed as a public API, and since we're rapidly moving standard libraries to gems, it would not be appropriate to use an internal API. So, we can either make WeakMap an official part of the public standard API, or do alternate 2.

Alternate 2: Add weak reference queues to the weakref API, so users can implement their own efficient weak maps. Some of this has been discussed (at great length) in https://bugs.ruby-lang.org/issues/4168, and the JRuby team has supported the weaklink gem for many years (which provides a WeakRef+RefQueue implementation for JRuby).

The original patch works well for small numbers of remoted objects.

Associated revisions

Revision 68e3f819 - 04/24/2019 04:08 PM - Masatoshi SEKI

add DRbObject dereference test (Preparation for investigation of Bug #15711)

Revision 8980b53a - 05/04/2019 10:28 AM - Masatoshi SEKI

add DRb::WeakIdConv (Bug #15711)

History

#1 - 03/19/2019 04:31 PM - headius (Charles Nutter)

We almost had agreement on adding a reference queue here: https://bugs.ruby-lang.org/issues/6309

#2 - 03/19/2019 04:47 PM - headius (Charles Nutter)

I added a mutex and pushed the base implementation as a PR here: https://github.com/ruby/ruby/pull/2102

I can try to work up patches for the other two implementations, but they'll need some additional work within MRI to support.

#3 - 03/19/2019 05:12 PM - headius (Charles Nutter)

Ok, unfortunately I'm not sure that WeakMap will work for this purpose. It works based on identity (which would not work in implementations where Fixnum-ranged Integers are not guaranteed to be the same object every time), attempts to attach a finalizer to its keys (which will not work for Integer-based keys anyway), and it can't work with frozen keys (so we can't use a symbol key).

Errors I got attempting to use it with simple IDs and with symbols generated based on those ID values:

```
  1) Error:
DRbTests::TestDRbAry#test_02_collect:
DRb::DRbConnError: cannot define finalizer for Integer
    /Users/headius/projects/ruby/lib/drb/drb.rb:390:in `[]='

DRbTests::TestDRbAry#test_02_collect:
DRb::DRbConnError: can't modify frozen Symbol
    /Users/headius/projects/ruby/lib/drb/drb.rb:390:in `[]='
```

Unless WeakMap is made into a general-purpose weak hash that supports any keys (without identity comparison) I'm not sure it will be a good option for us.

Patch is simple enough but doesn't work:

```
commit 416a4638b1043e62cd798bc364e3314d47f93fd8
Author: Charles Oliver Nutter <headius@headius.com>
Date:   Tue Mar 19 11:13:26 2019 -0500

    Replace use of _id2ref in DRb with a mapping based on WeakMap.

diff --git a/lib/drb/drb.rb b/lib/drb/drb.rb
index de57362f24..7300d9a53d 100644
--- a/lib/drb/drb.rb
+++ b/lib/drb/drb.rb
@@ -355,21 +355,25 @@ class DRbConnError < DRbError; end

   # Class responsible for converting between an object and its id.
   #
-  # This, the default implementation, uses an object's local ObjectSpace
+  # This, the default implementation, uses an object's runtime-assigned
   # __id__ as its id.  This means that an object's identification over
   # drb remains valid only while that object instance remains alive
   # within the server runtime.
   #
   # For alternative mechanisms, see DRb::TimerIdConv in drb/timeridconv.rb
   # and DRbNameIdConv in sample/name.rb in the full drb distribution.
+  #
   class DRbIdConv
+    def initialize
+      @id2ref = ObjectSpace::WeakMap.new
+    end

     # Convert an object reference id to an object.
     #
     # This implementation looks up the reference id in the local object
     # space and returns the object it refers to.
     def to_obj(ref)
-      ObjectSpace._id2ref(ref)
+      @id2ref[ref]
     end

     # Convert an object into a reference id.
@@ -377,7 +381,14 @@ def to_obj(ref)
     # This implementation returns the object's __id__ in the local
     # object space.
     def to_id(obj)
-      obj.nil? ? nil : obj.__id__
+      if obj.nil?
+        return nil
+      end
+
+      id = obj.__id__
+      @id2ref[id] = obj
+
+      id
     end
   end
```

#4 - 03/20/2019 12:50 AM - hsbt (Hiroshi SHIBATA)

- Assignee set to seki (Masatoshi Seki)

- Status changed from Open to Assigned

[seki (Masatoshi Seki)](#)

Can you handle this?

#5 - 03/20/2019 12:21 PM - Eregon (Benoit Daloze)

[headius (Charles Nutter)](#) Could you attach a diff for the weakref-based approach?
It seems the GitHub link doesn't work well (it shows the list of commits for me).

#6 - 03/26/2019 05:37 PM - headius (Charles Nutter)

- Assignee deleted (seki (Masatoshi Seki))

[Eregon (Benoit Daloze)](#) Are you talking about the PR? That has the version of code that should be merged into CRuby. The link in the description was just to show the diff in place in our CRuby fork, but it didn't base that diff against a moment-in-time snapshot of CRuby.

https://github.com/ruby/ruby/pull/2102/files

#7 - 03/26/2019 06:01 PM - headius (Charles Nutter)

I have not written up a patch based on a reference queue, but it would basically just use the IdWeakRef class and the _cleanup method from the IdHash example collection in my "weakling" library:

https://github.com/headius/weakling/blob/master/lib/weakling/collections.rb

This implementation polls the reference queue on each "clean", which will be a no-op when there are no vacated references in the queue. Compare to the full Hash scan in the supported weakref version. In order to use the more efficient impl, the weakref library would need to support reference queues.

#8 - 03/26/2019 06:13 PM - headius (Charles Nutter)

PR for the reference queue version is here: https://github.com/ruby/ruby/pull/2104

Won't pass CI because of the missing weakling dependency, but should be easy to test locally.

#9 - 03/27/2019 01:24 PM - Eregon (Benoit Daloze)

headius (Charles Nutter) wrote:

> [Eregon (Benoit Daloze)](#) Are you talking about the PR? That has the version of code that should be merged into CRuby. The link in the
> description was just to show the diff in place in our CRuby fork, but it didn't base that diff against a moment-in-time snapshot of CRuby.

Yes, that link shows the list of commits rather than the diff for some reason.

> https://github.com/ruby/ruby/pull/2102/files

Thanks, that's much easier to review.

[seki (Masatoshi Seki)](#) Could you review that PR?

#10 - 03/28/2019 12:30 AM - hsbt (Hiroshi SHIBATA)

- Assignee set to seki (Masatoshi Seki)

#11 - 03/30/2019 12:40 AM - headius (Charles Nutter)

I'm happy to chat/review this code and the overall _id2ref change at RubyKaigi in a few weeks, btw.

#12 - 05/04/2019 10:29 AM - Anonymous

- Status changed from Assigned to Closed

Applied in changeset [git|8980b53a48b1f55e09c5223008225e6bfa765405](#).

---

add DRb::WeakIdConv (Bug [#15711](#))