This ticket describes the semantics of "shareable" and proposes a new method `Ractor.make_shareable(obj)`.

With this method, `obj` becomes a shareable object by freezing it and reachable objects if it is necessary and it is possible.

**Background**

"Shareable object" is new term used by Ractors.

- (1) We can send a reference to send a shareable object instead of doing deep copy.
- (2) We can access to a constant which contains a shareable object from non-main ractors.

(1) is (mainly) performance and (2) is programmability (how to rewrite the libraries and so on). See [Feature #1727] for the examples of (2).

The definition of shareable object is thread-safe, ractor-safe object, they are safe to access from multiple ractors simultaneously.

The following conditions are definition of "shareable object" (`obj` is shareable object if ...).

- SPECIAL_CONST objects are shareable (also be frozen).
- if `RBASIC(obj)->flags | FL_SHAREABLE` is true, it is shareable.
- T_OBJECT: if all instance variables only refer to shareable objects (def1) and itself is frozen (def2)
- T_ARRAY: (def1) + (def2) + if all elements are shareable objects
- T_HASH: (def1) + (def2) + if all keys and values are shareable objects and default_proc/value (IFNONE, in C-level) is a sharable object
- T_STRUCT: (def1) + (def2) + if all members are shareable objects
- T_RATIONAL: (def1) + (def2) + if num/den are shareable
- T_COMPLEXL: (def1) + (def2) + if imag/real are shareable
- T_STRING, T_FILE, T_MATCH, T_REGEXP: (def1) + (def2)

T_DATA (user customizable data structure) is difficult problem because if it is frozen, it can modify a state (== we can use (def2)), for example current Queue implementation ignores frozen flag. So we define the semantics like:

- T_DATA: (def1) + if `RTYPEDDATA_P(obj)` is true and `rb_data_type_t::flags | RUBY_TYPED_FROZEN_SHAREABLE`, we rely on (def2). Otherwise, this T_DATA object can not become a shareable object. Also we need to check reachable objects are shareable.

`Ractor.shareable?(obj)` checks this definitions.

Note that you can add FL_SHAREABLE flag to any objects, so if you know there is no mutation or enough protected, you can set the flag and it will be a shareable object. For example, [Feature #17261](#17261) use this flag and TVars are shareable objects.

**Proposal**

As you can see, most of objects are shareable if they are frozen and they are only refers shareable/frozen objects. `Ractor.make_shareable(obj)` tries to freeze objects recursively if it is non-shareable objects.

```ruby
# pseudo-code

def Ractor.make_shareable(obj)
  return obj if Ractor.shareable?(obj)

  obj.freeze

  if obj.is T_DATA and (obj.type.flags | RUBY_TYPED_FROZEN_SHAREABLE) == 0
```

03/19/2022
raise "can not make shareable object for ..."
end

obj.reachable_objects{|o|  
  Ractor.make_shareable(o)
}

# only refer to the shareable objects, so it can be a shareable.
obj.set! FL_SHAREBLE
end

If it raises an error in the middle of the process, half-baked state are remained.

begin
  Ractor.make_shareable [  a1 = [1, 2],
    Thread.new{},
  a2 = [3, 4]]
 rescue Ractor::Error
end

p Ractor.shareable?(a1) #=> true
p Ractor.shareable?(a2) #=> false

Implementation

https://github.com/ruby/ruby/pull/3678
and it was already merged to propose https://bugs.ruby-lang.org/issues/17273

<table>
<thead>
<tr>
<th>Related issues:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related to Ruby master - Feature #17145: Ractor-aware <code>Object#deep_freeze</code></td>
</tr>
<tr>
<td>Related to Ruby master - Feature #17273: shareable_constant_value pragma</td>
</tr>
</tbody>
</table>

History

#1 - 10/21/2020 06:49 AM - ko1 (Koichi Sasada)
- Description updated

#2 - 10/21/2020 06:46 PM - Eregon (Benoit Daloze)
- Related to Feature #17145: Ractor-aware `Object#deep_freeze` added

#3 - 10/21/2020 06:47 PM - Eregon (Benoit Daloze)
ko1 (Koichi Sasada) What's the difference with Object#deep_freeze(skip_shareable: false) from #17145?

#4 - 10/21/2020 11:00 PM - ko1 (Koichi Sasada)
Eregon (Benoit Daloze) wrote in #note-3:

  ko1 (Koichi Sasada) What's the difference with Object#deep_freeze(skip_shareable: false) from #17145?

Almost same.

- the ability is to focus skip_shareable: true.
- this ticket describes the detailed semantics of making sharable protocol, including T_DATA. We don't need to describe T_CLASS/T_MODULE/T_ICLASS because they are sharable by birth.

#5 - 10/25/2020 01:38 PM - Eregon (Benoit Daloze)
- Related to Feature #17273: shareable_constant_value pragma added

#6 - 12/19/2020 10:12 PM - marcandre (Marc-Andre Lafortune)
- Status changed from Open to Closed