Ruby - Bug #17510

enable constant cache on Ractors

01/04/2021 04:47 PM - ko1 (Koichi Sasada)

Status:	Closed		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:			
ruby -v:		**Backport:**	2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONE

Description

constant cache IC is accessed by non-atomic manner and there are thread-safety issues, so Ruby 3.0 disables to use const cache on non-main ractors.

To enable the constant cache, we need to make IC access atomically, and this is a patch: https://github.com/ruby/ruby/pull/4022

This patch enables it by introducing imemo_constcache and allocates it by every re-fill of const cache like imemo_callcache.

Benchmark:

```
Warning[:experimental] = false

def tarai(x, y, z) = Object && # useless constant reference
  x <= y ? y : tarai(tarai(x-1, y, z),
                     tarai(y-1, z, x),
                     tarai(z-1, x, y))
def task = tarai(12, 6, 0)
#
require 'benchmark'

Benchmark.bm{|x|
  x.report('seq'){ 4.times{ task } }
  x.report('par'){ 4.times.map{ Ractor.new{ task } }.each(&:take) }
}

master (095972e799):
       user     system      total        real
seq  1.618664   0.001916   1.620580 (  1.620620)
par 18.168252  16.840400  35.008652 (  9.452195)

This patch:
       user     system      total        real
seq  1.587542   0.000000   1.587542 (  1.587548)
par  1.950062   0.000000   1.950062 (  0.497173)
```

Associated revisions

Revision e7fc353f044f9280222ca41b029b1368d2bf2fe3 - 01/04/2021 05:27 PM - ko1 (Koichi Sasada)

enable constant cache on ractors

constant cache IC is accessed by non-atomic manner and there are thread-safety issues, so Ruby 3.0 disables to use const cache on non-main ractors.

This patch enables it by introducing imemo_constcache and allocates it by every re-fill of const cache like imemo_callcache.
[Bug #17510]

Now IC only has one entry IC::entry and it points to iseq_inline_constant_cache_entry, managed by T_IMEMO object.

IC is atomic data structure so rb_mjit_before_vm_ic_update() and
rb_mjit_after_vm_ic_update() is not needed.

Revision e7fc353f044f9280222ca41b029b1368d2bf2fe3 - 01/04/2021 05:27 PM - ko1 (Koichi Sasada)

enable constant cache on ractors

constant cache IC is accessed by non-atomic manner and there are
thread-safety issues, so Ruby 3.0 disables to use const cache on
non-main ractors.

This patch enables it by introducing imemo_constcache and allocates
it by every re-fill of const cache like imemo_callcache.
[Bug #17510]

Now IC only has one entry IC::entry and it points to
iseq_inline_constant_cache_entry, managed by T_IMEMO object.

IC is atomic data structure so rb_mjit_before_vm_ic_update() and
rb_mjit_after_vm_ic_update() is not needed.

Revision e7fc353f - 01/04/2021 05:27 PM - ko1 (Koichi Sasada)

enable constant cache on ractors

constant cache IC is accessed by non-atomic manner and there are
thread-safety issues, so Ruby 3.0 disables to use const cache on
non-main ractors.

This patch enables it by introducing imemo_constcache and allocates
it by every re-fill of const cache like imemo_callcache.
[Bug #17510]

Now IC only has one entry IC::entry and it points to
iseq_inline_constant_cache_entry, managed by T_IMEMO object.

IC is atomic data structure so rb_mjit_before_vm_ic_update() and
rb_mjit_after_vm_ic_update() is not needed.

Revision b93e16dc0f45069d4a5fcce20d5c4437e151f0a8 - 01/13/2021 08:06 AM - ko1 (Koichi Sasada)

enable constant cache on ractors

constant cache IC is accessed by non-atomic manner and there are
thread-safety issues, so Ruby 3.0 disables to use const cache on
non-main ractors.

This patch enables it by introducing imemo_constcache and allocates
it by every re-fill of const cache like imemo_callcache.
[Bug #17510]

Now IC only has one entry IC::entry and it points to
iseq_inline_constant_cache_entry, managed by T_IMEMO object.

IC is atomic data structure so rb_mjit_before_vm_ic_update() and
rb_mjit_after_vm_ic_update() is not needed.

Revision b93e16dc0f45069d4a5fcce20d5c4437e151f0a8 - 01/13/2021 08:06 AM - ko1 (Koichi Sasada)

enable constant cache on ractors

constant cache IC is accessed by non-atomic manner and there are
thread-safety issues, so Ruby 3.0 disables to use const cache on
non-main ractors.

This patch enables it by introducing imemo_constcache and allocates
it by every re-fill of const cache like imemo_callcache.
[Bug #17510]

Now IC only has one entry IC::entry and it points to
iseq_inline_constant_cache_entry, managed by T_IMEMO object.

IC is atomic data structure so rb_mjit_before_vm_ic_update() and
rb_mjit_after_vm_ic_update() is not needed.

Revision b93e16dc - 01/13/2021 08:06 AM - ko1 (Koichi Sasada)

enable constant cache on ractors

constant cache IC is accessed by non-atomic manner and there are thread-safety issues, so Ruby 3.0 disables to use const cache on non-main ractors.

This patch enables it by introducing imemo_constcache and allocates it by every re-fill of const cache like imemo_callcache.
[Bug #17510]

Now IC only has one entry IC::entry and it points to iseq_inline_constant_cache_entry, managed by T_IMEMO object.

IC is atomic data structure so rb_mjit_before_vm_ic_update() and rb_mjit_after_vm_ic_update() is not needed.

History

#1 - 01/04/2021 05:28 PM - ko1 (Koichi Sasada)

- Status changed from Open to Closed

Applied in changeset git|e7fc353f044f9280222ca41b029b1368d2bf2fe3.

---

enable constant cache on ractors

constant cache IC is accessed by non-atomic manner and there are thread-safety issues, so Ruby 3.0 disables to use const cache on non-main ractors.

This patch enables it by introducing imemo_constcache and allocates it by every re-fill of const cache like imemo_callcache.
[Bug #17510]

Now IC only has one entry IC::entry and it points to iseq_inline_constant_cache_entry, managed by T_IMEMO object.

IC is atomic data structure so rb_mjit_before_vm_ic_update() and rb_mjit_after_vm_ic_update() is not needed.

#2 - 01/04/2021 07:09 PM - ko1 (Koichi Sasada)

- Backport changed from 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN to 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: REQUIRED

this patch is for performance, but it seems be valuable to backport to 3.0.1.

#3 - 01/04/2021 09:14 PM - k0kubun (Takashi Kokubun)

e7fc353f04 broke MJIT, so if it's to be backported to 3.0.1, please make sure (87c546b5fa to avoid a conflict and) 7a3322a0fd is backported as well.

#4 - 01/13/2021 04:34 AM - naruse (Yui NARUSE)

- Backport changed from 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: REQUIRED to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONTNEED, 3.0: REQUIRED

#5 - 02/01/2021 09:23 AM - naruse (Yui NARUSE)

- Backport changed from 2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONTNEED, 3.0: REQUIRED to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONTNEED, 3.0: DONE

ruby_3_0 b93e16dc0f45069d4a5fcce20d5c4437e151f0a8.