

Ruby - Bug #15050

GC after forking with fibers crashes

08/30/2018 07:54 PM - normalperson (Eric Wong)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
ruby -v:		
		Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
Description		
Not sure when to work on this, so leaving this here for now...		
The management of stacks for root fiber and regular fibers is different and confusing. Perhaps unifying thread and fiber stack cache is the best way to go.		
Is a separate class of stacks even necessary? We should aim to minimize use of stack space.		
"[BUG] Illegal root fiber parameter"		

Associated revisions

Revision c99b9eb03952284f213b714acaf7f4885ce177c2 - 09/12/2018 08:49 PM - Eric Wong

share VM stack between threads and fibers if identical in size

ec->vm_stack is always allocated with malloc, so stack cache for root fiber (thread stack) and non-root fibers can be shared as long as the size is the same. The purpose of this change is to reduce dependencies on ROOT_FIBER_CONTEXT.

[Feature #15095] [Bug #15050]

v2: vm.c: fix build with USE_THREAD_DATA_RECYCLE==0

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64703 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c99b9eb03952284f213b714acaf7f4885ce177c2 - 09/12/2018 08:49 PM - Eric Wong

share VM stack between threads and fibers if identical in size

ec->vm_stack is always allocated with malloc, so stack cache for root fiber (thread stack) and non-root fibers can be shared as long as the size is the same. The purpose of this change is to reduce dependencies on ROOT_FIBER_CONTEXT.

[Feature #15095] [Bug #15050]

v2: vm.c: fix build with USE_THREAD_DATA_RECYCLE==0

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64703 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision c99b9eb0 - 09/12/2018 08:49 PM - Eric Wong

share VM stack between threads and fibers if identical in size

ec->vm_stack is always allocated with malloc, so stack cache for root fiber (thread stack) and non-root fibers can be shared as long as the size is the same. The purpose of this change is to reduce dependencies on ROOT_FIBER_CONTEXT.

[Feature #15095] [Bug #15050]

v2: vm.c: fix build with USE_THREAD_DATA_RECYCLE==0

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64703 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d40694de726aab80ced4d4a264d08664db2112d2 - 09/12/2018 08:49 PM - Eric Wong

cont.c (fiber_memsizes): do not rely on ROOT_FIBER_CONTEXT

We can check if the fiber we're interested in is the
th->root_fiber for the owner thread, so there is no need to use
ROOT_FIBER_CONTEXT.

Note: there is no guarantee th->ec points to
&th->root_fiber->cont.saved_ec, thus vm::thread_memsizes may not
account for root fiber correctly (pre-existing bug).

[Bug #15050]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64705 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d40694de726aab80ced4d4a264d08664db2112d2 - 09/12/2018 08:49 PM - Eric Wong

cont.c (fiber_memsizes): do not rely on ROOT_FIBER_CONTEXT

We can check if the fiber we're interested in is the
th->root_fiber for the owner thread, so there is no need to use
ROOT_FIBER_CONTEXT.

Note: there is no guarantee th->ec points to
&th->root_fiber->cont.saved_ec, thus vm::thread_memsizes may not
account for root fiber correctly (pre-existing bug).

[Bug #15050]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64705 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d40694de - 09/12/2018 08:49 PM - Eric Wong

cont.c (fiber_memsizes): do not rely on ROOT_FIBER_CONTEXT

We can check if the fiber we're interested in is the
th->root_fiber for the owner thread, so there is no need to use
ROOT_FIBER_CONTEXT.

Note: there is no guarantee th->ec points to
&th->root_fiber->cont.saved_ec, thus vm::thread_memsizes may not
account for root fiber correctly (pre-existing bug).

[Bug #15050]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64705 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 12409ad28cf909f84ccae8096bba2beb442266df - 09/12/2018 08:49 PM - Eric Wong

fiber: fix crash on GC after forking

Remove the remainder of ROOT_FIBER_CONTEXT use and unnecessary
differences between the root and non-root fiber. This makes
it easier to follow new root fiber at fork time.

Multiple sources of truth often leads to bugs, as in this case.
We can determine root fiber by checking a fiber against the root_fiber
of its owner thread. The new 'fiber_is_root_p' function
supports that.

Now, we can care only about free-ing/recycling/munmap-ing stacks
as appropriate.

[Bug #15050]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64706 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 12409ad28cf909f84ccae8096bba2beb442266df - 09/12/2018 08:49 PM - Eric Wong

fiber: fix crash on GC after forking

Remove the remainder of ROOT_FIBER_CONTEXT use and unnecessary

differences between the root and non-root fiber. This makes it easier to follow new root fiber at fork time.

Multiple sources of truth often leads to bugs, as in this case. We can determine root fiber by checking a fiber against the root_fiber of its owner thread. The new 'fiber_is_root_p' function supports that.

Now, we can care only about free-ing/recycling/munmap-ing stacks as appropriate.

[Bug #15050]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64706 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 12409ad2 - 09/12/2018 08:49 PM - Eric Wong

fiber: fix crash on GC after forking

Remove the remainder of ROOT_FIBER_CONTEXT use and unnecessary differences between the root and non-root fiber. This makes it easier to follow new root fiber at fork time.

Multiple sources of truth often leads to bugs, as in this case. We can determine root fiber by checking a fiber against the root_fiber of its owner thread. The new 'fiber_is_root_p' function supports that.

Now, we can care only about free-ing/recycling/munmap-ing stacks as appropriate.

[Bug #15050]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64706 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 08/30/2018 08:03 PM - normalperson (Eric Wong)

normalperson@yhbt.net wrote:

<https://bugs.ruby-lang.org/issues/15050>

This fixes the immediate bug, but I think there's other problems with stack management

```
diff --git a/cont.c b/cont.c
index 7bc1724176..ab42dfb27a 100644
--- a/cont.c
+++ b/cont.c
@@ -1983,6 +1983,7 @@ rb_fiber_atfork(rb_thread_t *th)
 {
   if (th->root_fiber) {
     if (&th->root_fiber->cont.saved_ec != th->ec) {
       +      th->root_fiber->cont.type = FIBER_CONTEXT;
       th->root_fiber = th->ec->fiber_ptr;
       th->root_fiber->cont.type = ROOT_FIBER_CONTEXT;
     }
   }
```

#2 - 08/31/2018 03:01 AM - ko1 (Koichi Sasada)

Is this because [Bug #15041]?

#3 - 08/31/2018 03:32 AM - normalperson (Eric Wong)

ko1@atdot.net wrote:

Is this because [Bug #15041]?

Yes, fix one bug, hit another :< Story of my life

(I found these while working on auto-fiber)

#4 - 08/31/2018 03:52 AM - normalperson (Eric Wong)

Koichi Sasada ko1@atdot.net wrote:

On 2018/08/31 12:30, Eric Wong wrote:

Yes, fix one bug, hit another :< Story of my life

[Bug [#15041](#)] hits something wrong?
(sorry I needed to ask earlier)

No, it's not wrong; but it is an incomplete fix. There seems to be several problems related to fork + Fiber

#5 - 08/31/2018 04:03 AM - ko1 (Koichi Sasada)

On 2018/08/31 12:50, Eric Wong wrote:

[Bug [#15041](#)] hits something wrong?
(sorry I needed to ask earlier)

No, it's not wrong; but it is an incomplete fix. There seems to be several problems related to fork + Fiber

Sorry my question is wrong.

What is the problem [Bug [#15041](#)] want to solve?

I remember that there are several implicit assumption on root fiber and so on. I think changing this attribute is bad idea now.

--
// SASADA Koichi at atdot dot net

#6 - 08/31/2018 06:22 AM - normalperson (Eric Wong)

Koichi Sasada ko1@atdot.net wrote:

On 2018/08/31 12:50, Eric Wong wrote:

[Bug [#15041](#)] hits something wrong?
(sorry I needed to ask earlier)

No, it's not wrong; but it is an incomplete fix. There seems to be several problems related to fork + Fiber

Sorry my question is wrong.

What is the problem [Bug [#15041](#)] want to solve?

Switching fiber can crash in child process. r64589 test change shows it:

```
- Fiber.new{ pid = fork {} }.resume
+ Fiber.new do
+   pid = fork do
+     Fiber.new {}.transfer
+   end
+ end.resume
```

I remember that there are several implicit assumption on root fiber and so on. I think changing this attribute is bad idea now.

I think we need to change to avoid crashes. We change vm->main_thread at fork, too. Maybe I can investigate late next week.

#7 - 08/31/2018 07:22 AM - ko1 (Koichi Sasada)

On 2018/08/31 15:14, Eric Wong wrote:

What is the problem [Bug #15041] want to solve?

Switching fiber can crash in child process. r64589 test change shows it:

```
-      Fiber.new{ pid = fork {} }.resume
+      Fiber.new do
+        pid = fork do
+          Fiber.new {}.transfer
+        end
+      end.resume
```

I remember that there are several implicit assumption on root fiber and so on. I think changing this attribute is bad idea now.

I think we need to change to avoid crashes. We change vm->main_thread at fork, too. Maybe I can investigate late next week.

Ok. We need to avoid this kind of crash.

--
// SASADA Koichi at atdot dot net

#8 - 09/09/2018 10:12 AM - normalperson (Eric Wong)

Koichi Sasada ko1@atdot.net wrote:

Ok. We need to avoid this kind of crash.

<https://bugs.ruby-lang.org/issues/15050>

OK, I think I fixed the issue (also including [Feature #15095])

Greg, there's a small win32-specific change in this series which seems low risk, but I figured I'd Cc you anyways for testing. Thanks.

Pull request:

The following changes since commit 64b326c2204e562aa3d6025ca097a82bcf4de303:

spec/ruby/library/socket/addinfo: require for SocketSpecs (2018-09-09 08:50:53 +0000)

are available in the Git repository at:

<https://80x24.org/ruby.git> fiber-fork

for you to fetch changes up to ce0a6005fbf78203b3bcaa4b90bf94b8b5c44782:

fiber: fix crash on GC after forking (2018-09-09 09:49:55 +0000)

Or broken out patches:

- [PATCH 1/4] share VM stack between threads and fibers if identical
<https://80x24.org/spew/20180909095347.30757-2-e@80x24.org/raw>
- [PATCH 2/4] cont.c (ec_set_vm_stack): avoid needless casting
<https://80x24.org/spew/20180909095347.30757-3-e@80x24.org/raw>
- [PATCH 3/4] cont.c (fiber_memsize): do not rely on ROOT_FIBER_CONTEXT
<https://80x24.org/spew/20180909095347.30757-4-e@80x24.org/raw>
- [PATCH 4/4] fiber: fix crash on GC after forking
<https://80x24.org/spew/20180909095347.30757-5-e@80x24.org/raw>

Or full diff against current trunk (r64663):

<https://80x24.org/spew/20180909100319.ild2rlncrmmm2lwp@whir/raw>

#9 - 09/12/2018 08:49 PM - normalperson (Eric Wong)

- Status changed from Open to Closed

Applied in changeset trunk|r64703.

share VM stack between threads and fibers if identical in size

ec->vm_stack is always allocated with malloc, so stack cache for root fiber (thread stack) and non-root fibers can be shared as long as the size is the same. The purpose of this change is to reduce dependencies on ROOT_FIBER_CONTEXT.

[Feature [#15095](#)] [Bug [#15050](#)]

v2: vm.c: fix build with USE_THREAD_DATA_RECYCLE==0

#10 - 09/12/2018 10:32 PM - MSP-Greg (Greg L)

[@normalperson \(Eric Wong\)](#)

Eric,

Sorry been busy with 'this is a bigger mess than I thought' kinds of things.

I just ran ruby-loco on r64706 'fiber: fix crash on GC after forking', and the build passed.

Thanks, Greg

Files

0001-test_fiber-show-crash-on-GC-after-forking.patch	1.19 KB	08/30/2018	normalperson (Eric Wong)
--	---------	------------	--------------------------