Ruby master - Bug #10892

Deadlock in autoload

02/23/2015 12:41 PM - Eregon (Benoit Daloze)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.3.0dev (2015-02-23 trunk 49693) [x86_64-linux]	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

Updating to recent RubySpec seems to show a bug under concurrent autoload.
I attach the extracted logic to reproduce.

At me, the script ends with either, in most cases,

```
autoload_bug.rb:105:in `value': No live threads left. Deadlock? (fatal)
    from autoload_bug.rb:105:in `map'
    from autoload_bug.rb:105:in `<main>'
```

Or:

```
autoload_bug.rb:95:in `const_get': uninitialized constant Mod1 (NameError)
    from autoload_bug.rb:95:in `block (3 levels) in <main>'
    from autoload_bug.rb:86:in `each'
    from autoload_bug.rb:86:in `block (2 levels) in <main>'
```

Which both seem incorrect behavior.
All versions from 2.0 seem affected, and 1.9.3 behavior seems wrong but differently.

Could someone confirm this is a bug?
Is it likely to be fixed?

Related issues:

Related to Ruby master - Bug #7530: Concurrent loads fail with mutex errors	**Closed**

History

#1 - 07/31/2015 04:59 AM - thedarkone (Vit Z)

- File 0001-load.c-unlock-the-new-shield.patch added

That broken rubyspec was written by me. The problem lies with repeatedly autoloading the same .rb file, since this should be impossible, the spec manually deletes the loaded path from $LOADED_FEATURES and then re-declares the autoload, this is currently broken on MRI.

Here's a much smaller repro script:

```
def with_autoload_file(const_name, file_name = 'foo.rb')
  mangled_file_name = file_name.sub(/\.rb\Z/, '____temp____autoload.rb')
# avoid accidentally overwriting any files
  File.write(mangled_file_name, "sleep 1; module #{const_name}; end")
  autoload const_name, File.expand_path(mangled_file_name.sub(/\.rb\Z/, ''))
  $LOADED_FEATURES.delete(File.expand_path(mangled_file_name)) if $LOADED_FEATURES.include?(File.expand_path(
mangled_file_name))
  yield
ensure
  File.delete(mangled_file_name)
end

foo_ready = bar_waiting = bar_ready = false
t = Thread.new do
  Thread.pass until foo_ready
  Foo
  bar_waiting = true
  Thread.pass until bar_ready
  Bar
```

```
end

with_autoload_file('Foo') do
  foo_ready = true
  Foo
end

Thread.pass until bar_waiting

with_autoload_file('Bar') do
  bar_ready = true
  Bar
end

t.join
```

Running this results in an "uninitialized constant Bar" exception from the non-main thread.

If the last block is rearranged like this:

```
with_autoload_file('Bar') do
  Bar
  bar_ready = true
end
```

the script deadlocks (main thread deadlocks, while secondary thread t busy spins in Thread.pass until bar_ready).

If the last autoload block uses a different .rb file, everything works fine:

```
with_autoload_file('Bar', 'bar.rb') do
  Bar
  bar_ready = true
end
```

I think I've tracked the issue to an incorrectly locked load_lock's thread_shield: when rb_thread_shield_wait() returns Qfalse the failed thread creates a new thread_shield via rb_thread_shield_new(), however because rb_thread_shield_new() automatically locks the newly created shield and the branch does not return a successful ftptr, the newly installed shield is then never unlocked.

The attached patch seems to fix the issue for me.

#2 - 07/31/2015 05:59 AM - nobu (Nobuyoshi Nakada)

- Related to Bug #7530: Concurrent loads fail with mutex errors added

#3 - 10/12/2015 01:22 PM - Eregon (Benoit Daloze)

Could someone review the patch and apply it or find an alternative fix?

#4 - 10/14/2015 07:58 PM - normalperson (Eric Wong)

eregontp@gmail.com wrote:

> Could someone review the patch and apply it or find an alternative fix?

Fwiw, I mentioned in [ruby-core:70359] that I tried it for [Bug #11384]
without success, but Redmine + list integration was broken at the
time.

#5 - 10/15/2015 11:48 AM - Eregon (Benoit Daloze)

On Wed, Oct 14, 2015 at 9:56 PM, Eric Wong normalperson@yhbt.net wrote:

> Fwiw, I mentioned in [ruby-core:70359] that I tried it for [Bug #11384]
> without success, but Redmine + list integration was broken at the
> time.

Ah indeed I missed that, thanks.
Did you try for this issue in particular?

About #11384, I guess we need another fix then :/

#6 - 10/16/2015 01:58 AM - normalperson (Eric Wong)

Benoit Daloze eregontp@gmail.com wrote:

> On Wed, Oct 14, 2015 at 9:56 PM, Eric Wong normalperson@yhbt.net wrote:
>
> > Fwiw, I mentioned in [ruby-core:70359] that I tried it for [Bug #11384]
> > without success, but Redmine + list integration was broken at the
> > time.
>
> Ah indeed I missed that, thanks.
> Did you try for this issue in particular?

Ah, yes, the repro script in [ruby-core:70197] does get fixed on my
machine with the patch. I don't understand this code enough to
know if it breaks anything else, or if #11384 is a different bug
or a different manifestation of the same bug.

#7 - 05/08/2018 01:27 AM - eugeneius (Eugene Kenny)

The simpler repro script runs successfully from 2.3.0 onwards, and git bisect between 2.2.0 and 2.3.0 shows that r59221 (from #11384) fixed it.

#8 - 07/08/2019 01:03 AM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

Files

autoload_bug.rb	2.18 KB	02/23/2015	Eregon (Benoit Daloze)
0001-load.c-unlock-the-new-shield.patch	1005 Bytes	07/31/2015	thedarkone (Vit Z)