Backport192 - Backport #3840

Ruby 1.9.2p0 crashes using Rails 3.0 on repeated browser refresh

09/16/2010 09:39 PM - s_p_oneil (Sean O'Neil)

Status:	Rejected	
Priority:	Normal	
Assignee:	yugui (Yuki Sonoda)	

Description

=begin
I'm not certain whether this is a Ruby issue or a Rails issue (or a bit of both). The problem is intermittent, and it appears to be a threading issue surrounding the reading of the Rails log file. It is fairly easy for me to reproduce the problem:

1) Install a clean 1.9.2p0 and run "gem install rails" to get Rails 3.0.
2) Use the Rails 3.0 generator script to create an empty project.
3) Launch the web app using default options.
4) Point a browser to the home page for the new Rails app, and keep pressing F5 (or Refresh) at varying speeds until Ruby crashes. It seems to happen most often if I hold F5 for a few seconds and then release for a few seconds, varying how long I hold and release it.

It crashes eventually no matter how I run it, but it seems to crash more often in production mode (which logs a bunch of errors using the default generated app). The crash appears to be dependent on timing, but I can usually make it crash in less than 60 seconds.

I'm currently using Windows 7 (x64) and compiling Ruby with Visual Studio 2008. However, the crash also occurs with the pre-packaged RubyInstaller for Windows. Given the debug information I've provided below, it does not appear to be platform-specific, but a different OS can have different IO and thread timing issues, so it may be harder to reproduce on other platforms.

I rebuilt Ruby with debugging options and got this call stack when it crashed:
rb_econv_convert() in transcode.c line 1449 (the ec parameter was set to NULL)
fill_cbuf() in io.c line 1691
read_all() in io.c line 1755
io_read() in io.c line 2163

I added some additional checks to the code to find out how fptr->readconv was getting cleared inside fill_cbuf(), and I traced it to io_fillbuf() and then rb_read_internal(). Now, rb_read_internal() does not have a pointer to fptr or any of its members. It could be a buffer overrun, but the buffer is much larger than the file, and the changes seem much too clean for that. In addition to readconv being cleared, other struct members are being reset to an earlier state (not necessarily back to 0), which seems to imply that another thread is accessing the same IO object while the current thread is inside rb_read_internal(). I can't "catch it in the act" because setting breakpoints affects the timing, and it typically requires hundreds of passes through this code to break. However, I can copy the struct fptr is pointing to and have it break as soon as those members change. The info being read right before the IO object is reset is:

rbuf   0x05700408 "
Started GET "/javascripts/dragdrop.js?1283353508" for 127.0.0.1 at 2010-09-08 16:22:14 -0400
ActionController::RoutingError (No route matches "/javascripts/dragdrop.js"):

Rendered
d:/Projects/NetAuditor/v3.00/Code/Build/lib/ruby/gems/1.9.1/gems/actionpack-3.0.0/lib/action_dispatch/middleware/templates/rescues/routing_error.erb within rescues/layout (2.0ms)

"

I can eliminate this crash by changing the Ruby source to check for NULL before referencing the pointer. However, it may only hide the issue, which may come back to bite me later. So, does this sound like a problem that needs to be fixed in the Ruby code, or a thread-safety issue in Rails?

Thank you,
Sean O'Neil
=end

Associated revisions

Revision 6effaa9a - 06/18/2011 03:00 PM - nobu (Nobuyoshi Nakada)

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32169 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 32169 - 06/18/2011 03:00 PM - nobu (Nobuyoshi Nakada)

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

Revision 32169 - 06/18/2011 03:00 PM - nobu (Nobuyoshi Nakada)

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

Revision 32169 - 06/18/2011 03:00 PM - nobu (Nobuyoshi Nakada)

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

Revision 32169 - 06/18/2011 03:00 PM - nobu (Nobuyoshi Nakada)

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

Revision 32169 - 06/18/2011 03:00 PM - nobu (Nobuyoshi Nakada)

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

Revision 32169 - 06/18/2011 03:00 PM - nobu (Nobuyoshi Nakada)

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

History

#1 - 10/24/2010 02:48 PM - nobu (Nobuyoshi Nakada)

=begin
Hi,

At Thu, 16 Sep 2010 21:40:03 +0900,
Sean O'Neil wrote in [ruby-core:32428]:

> I can eliminate this crash by changing the Ruby source to
> check for NULL before referencing the pointer. However, it
> may only hide the issue, which may come back to bite me
> later. So, does this sound like a problem that needs to be
> fixed in the Ruby code, or a thread-safety issue in Rails?

Your investigation seems correct, and I agree that it should be
noticed earlier. I think Rails has a thread-safety issue but
the crash is definitely a bug of Ruby itself. Does this patch
fix it?

diff --git a/io.c b/io.c
index 4b8d34a..67d97f2 100644
--- a/io.c
+++ b/io.c
@@ -1722,6 +1722,7 @@ fill_cbuf(rb_io_t *fptr, int ec_flags)
if (fptr->rbuf_len == 0) {
READ_CHECK(fptr);
if (io_fillbuf(fptr) == -1) {

- rb_io_check_closed(fptr); ds = dp = (unsigned char *)fptr->cbuf + fptr->cbuf_off + fptr->cbuf_len; de = (unsigned char *)fptr->cbuf + fptr->cbuf_capa; res = rb_econv_convert(fptr->readconv, NULL, NULL, &dp, de, 0);

--

Nobu Nakada

=end

#2 - 01/26/2011 02:50 AM - Papillon (Ingmar Stieger)

=begin
I have exactly the same problem as Sean, and the call to rb_io_check_closed(fptr) does not help. In my case, the crash happens on 2-3 page reloads, so almost instantly.

What does help is this crude fix:

--- a/transcode.c 2010-06-17 09:55:46 +0000
+++ b/transcode.c 2011-01-25 17:08:01 +0000
@@ -1446,6 +1446,12 @@
unsigned char empty_buf;
unsigned char *empty_ptr = &empty_buf;

- if (!ec)
- {
- rb_raise(rb_eUndefinedConversionError, "ec is null (ing)");
- return(econv_undefined_conversion);

- }
  +
  ec->started = 1;

  if (!input_ptr) {

This results in the following console message:

[2011-01-25 18:13:17] ERROR Encoding::UndefinedConversionError: ec is null (ing)
c:/Ruby192/lib/ruby/gems/1.9.1/gems/railties-3.0.1/lib/rails/rack/log_tailer.rb:23:in read'
c:/Ruby192/lib/ruby/gems/1.9.1/gems/railties-3.0.1/lib/rails/rack/log_tailer.rb:23:intail!'
c:/Ruby192/lib/ruby/gems/1.9.1/gems/railties-3.0.1/lib/rails/rack/log_tailer.rb:15:in call'
c:/Ruby192/lib/ruby/gems/1.9.1/gems/rack-1.2.1/lib/rack/content_length.rb:13:incall'
c:/Ruby192/lib/ruby/gems/1.9.1/gems/rack-1.2.1/lib/rack/handler/webrick.rb:52:in service'
c:/Ruby192/lib/ruby/1.9.1/webrick/httpserver.rb:111:inservice'
c:/Ruby192/lib/ruby/1.9.1/webrick/httpserver.rb:70:in run'
c:/Ruby192/lib/ruby/1.9.1/webrick/server.rb:183:inblock in start_thread'

At least it does not crash any more, though I am not sure what the side effects of this might be.

Any ideas for a real solution ?

Ingmar

=end

#3 - 04/21/2011 10:23 AM - h.shirosaki (Hiroshi Shirosaki)

=begin
I wrote a test code.

Ruby raises "ec is null" error.

ruby 1.9.2p180 (2011-02-18 revision 30909) [i386-mingw32]

#!/usr/bin/env ruby

path = "test.txt"

file = File.open(path, "w")
file.close

threads = []

file = File.open(path, "a")
cursor = File.size(path)
read_file = File.open(path, "r")

1000.times do |i|
t = Thread.new do

```
file.print(i)
file.flush
end
threads << t

t = Thread.new do
content = read_file.read
cursor += content.size
read_file.seek cursor
print content
end
threads << t
end

threads.each { |t| t.join }

file.close
read_file.close

=end
```

#4 - 04/22/2011 05:54 AM - stepheneb (Stephen Bannasch)

```
=begin
On Mac OS X 10.6.7 using  ruby 1.9.2p180 your test appears to run correctly.
=end
```

#5 - 04/22/2011 08:23 PM - iviney (Ian Viney)

```
=begin
I have been having the same problem with Rails (see ((())).  I tried an even simpler version of  Hiroshi Shirosaki's test, and this fails repeatedly within 20 iterations.  Usually, Ruby crashes with an exit code of -1073741819.

I am using Windows XP with Ruby 1.9.2-p180.  It appears that this is a Windows-specific problem.

path = "test.txt"

file = File.open(path, "w")
read_file = File.open(path, "r")

1000.times do |i|
puts i
Thread.new { file.print(i) }
Thread.new { read_file.read }
end

file.close
read_file.close

=end
```

#6 - 04/23/2011 10:56 PM - luislavena (Luis Lavena)

```
=begin
Interesting.

Waiting for the threads to complete their operations avoids the segfault:

path = "test.txt"

file = File.open(path, "a")
read_file = File.open(path, "r")

1000.times do |i|
puts i
a = Thread.new { file.print(i) }
b = Thread.new { read_file.read }
[a, b].each { |t| t.join }
end

file.close
read_file.close

=end
```

#7 - 04/28/2011 10:55 AM - h.shirosaki (Hiroshi Shirosaki)

=begin
Simpler version tests look good.
I tried, but Ruby didn't crash.

I am using Windows XP with Ruby 1.9.2-p180.

Modifying threads join, Ruby crashes with my environment.

```
path = "test.txt"

file = File.open(path, "w")
read_file = File.open(path, "r")

threads = []
1000.times do |i|
threads << Thread.new { file.print(i) }
threads << Thread.new { read_file.read }
end
threads.each { |t| t.join }

file.close
read_file.close
```

=end

#8 - 04/28/2011 11:34 AM - luislavena (Luis Lavena)

=begin
Hiroshi Shirosaki wrote:

> Simpler version tests look good.
> I tried, but Ruby didn't crash.
>
> I am using Windows XP with Ruby 1.9.2-p180.
>
> Modifying threads join, Ruby crashes with my environment.

It crash for me.

Now, something to point out is that multiple threads performing read operations over the same file descriptor without a mutex or a synchronization mechanism is set to fail.

Same for writing operations, these need to be queued/synchronized properly.

Threads are not warranted to be executed in sequence, so that could be one of the issues we are seeing here.

=end

#9 - 06/17/2011 08:15 PM - h.shirosaki (Hiroshi Shirosaki)

This fix works well for me.
It seems that other thread calls clear_readconv in read_all while io_fillbuf select waiting.

```
--- ruby-1.9.2-p180_org/io.c    2011-01-16 21:35:27 +0900
+++ ruby-1.9.2-p180/io.c    2011-06-17 19:26:15 +0900
@@ -1689,6 +1689,9 @@
if (fptr->rbuf_len == 0) {
READ_CHECK(fptr);
if (io_fillbuf(fptr) == -1) {
```

- if(!fptr->readconv) {
- return MORE_CHAR_FINISHED;
- } ds = dp = (unsigned char *)fptr->cbuf + fptr->cbuf_off + fptr->cbuf_len; de = (unsigned char *)fptr->cbuf + fptr->cbuf_capa; res = rb_econv_convert(fptr->readconv, NULL, NULL, &dp, de, 0);

#10 - 06/18/2011 11:17 PM - luislavena (Luis Lavena)

- Target version set to 1.9.2

Hiroshi Shirosaki wrote:

> This fix works well for me.

It seems that other thread calls clear_readconv in read_all while io_fillbuf select waiting.

I can't no longer reproduce this on trunk (r32151):

1.rb:10:in new': failed to allocate memory (NoMemoryError)
from 1.rb:10:inblock in '
from 1.rb:7:in times'
from 1.rb:7:in'

However, your patch seems to have solved the segfault.

#11 - 06/19/2011 12:00 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r32169.
Sean, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- io.c (fill_cbuf): finish reading at EOF, and the readconv has been cleared by another thread while io_fillbuf() is waiting at select(). a patch in [ruby-core:37197] by Hiroshi Shirosaki . fixed #3840

#12 - 09/22/2011 05:08 AM - cheezy (Jeff Morgan)

Is there any chance this fix can be backported to 1.9.2? Without this fix ruby crashes very frequently when running a simple rails application on windows. Thanks!

#13 - 09/22/2011 01:25 PM - usa (Usaku NAKAMURA)

- Status changed from Closed to Assigned

- Tracker changed from Bug to Backport

- Project changed from Ruby trunk to Backport192

- Assignee set to yugui (Yuki Sonoda)

- Target version deleted (1.9.2)

#14 - 05/30/2016 08:33 AM - naruse (Yui NARUSE)

- Status changed from Assigned to Rejected