Ruby master - Bug #15130
Nested Timeouts hang on cygwin
09/17/2018 04:31 AM - duerst (Martin Dürst)

Status: Closed
Priority: Normal
Assignee: normalperson (Eric Wong)
Target version: ruby -v:
ruby 2.6.0dev (2018-09-16 trunk 64761)
[x86_64-cygwin]
Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN

Description
After using make; make runnable, the following one-liner hangs:

```
./ruby -ropen-uri -e 'open "http://web.mit.edu/index.html"
```

This happens on any http URI, but not for local files. The above URI is just an example. I would have chosen ruby-lang.org if it was still using http, but I didn't want to complicate this with https.

The open work for local files. I discovered this when trying to use make up, where the downloading of Unicode data files just hung.

Here's the trace I get after hitting Ctrl-C:

Traceback (most recent call last):
  15: from -e:1:in <main>
  14: from /cygdrive/c/Data/ruby_test_copy/lib/open-uri.rb:35:in open
  11: from /cygdrive/c/Data/ruby_test_copy/lib/open-uri.rb:224:in open_loop
  8: from /cygdrive/c/Data/ruby_test_copy/lib/open-uri.rb:758:in buffer_open
  7: from /cygdrive/c/Data/ruby_test_copy/lib/open-uri.rb:337:in open_http
  6: from /cygdrive/c/Data/ruby_test_copy/lib/net/http.rb:924:in start
  5: from /cygdrive/c/Data/ruby_test_copy/lib/net/http.rb:935:in do_start
  4: from /cygdrive/c/Data/ruby_test_copy/lib/net/http.rb:950:in connect
  3: from /cygdrive/c/Data/ruby_test_copy/lib/timeout.rb:103:in timeout
  2: from /cygdrive/c/Data/ruby_test_copy/lib/timeout.rb:97:in block in timeout
  1: from /cygdrive/c/Data/ruby_test_copy/lib/timeout.rb:97:in ensure in block in timeout
/cygdrive/c/Data/ruby_test_copy/lib/timeout.rb:97:injoin": Interrupt

My plan is to (manually) bisect this to find the offending commit. One suspicion I have is that the problem is related to timeouts, but this is not confirmed at all. Any help is appreciated.

Related issues:
- Related to Ruby master - Bug #15072: thread.c:4356:5: error: implicit declaration of function ‘ubf_list_atfork’ added
- Assignee set to normalperson (Eric Wong)

History
#1 - 09/19/2018 06:06 AM - duerst (Martin Dürst)
- Related to Bug #15072: thread.c:4356:5: error: implicit declaration of function ‘ubf_list_atfork’ added

#2 - 09/20/2018 03:32 AM - duerst (Martin Dürst)
This issue doesn't exist before #15072 started at r64485 and made it impossible to check whether open-uri works or not. I'm assigning this to Eric (normalperson) because it may well be timeout-related. Sorry if that turns out not to be the case.

#3 - 09/20/2018 03:33 AM - duerst (Martin Dürst)
- Assignee set to normalperson (Eric Wong)

#4 - 09/20/2018 05:32 AM - normalperson (Eric Wong)
I think I know what's going on (but need to do offline stuff).
Should have a patch for you to test in a few hours.

5 - 09/21/2018 12:48 AM - normalperson (Eric Wong)
- File 0001-thread_pthread.c-fall-back-to-UBF_TIMER_PTHREAD-for-.patch added

Martin: can you try the attached patch? And please commit it if it works.

    thread_pthread.c: fail back to UBF_TIMER_PTHREAD for Cygwin

    UBF_TIMER_NONE is not useful until we can make all I/O
    non-blocking <https://bugs.ruby-lang.org/issues/14968>
    For now, UBF_TIMER_PTHREAD should be a safe bet even for
    systems (e.g. Cygwin) without ubf_list.

    [Bug #15130]

Thanks

6 - 09/21/2018 01:30 AM - duerst (Martin Dürst)

normalperson (Eric Wong) wrote:

    Martin: can you try the attached patch? And please commit it if it works.

Thanks a lot for your patch. I'm working on trying it out, but I will only have the results later in my (Japanese) afternoon, sorry.

7 - 09/21/2018 06:07 AM - duerst (Martin Dürst)

I applied the patch, but got the same problem: I had to use Ctrl-C, and got essentially the same backtrace:

Traceback (most recent call last):
  15: from -e:1:<main>'
  14: from /cygdrive/c/Data/ruby_test_patch_pthread/lib/open-uri.rb:35:inopen'
  11: from /cygdrive/c/Data/ruby_test_patch_pthread/lib/open-uri.rb:224:in open_loop'
  10: from /cygdrive/c/Data/ruby_test_patch_pthread/lib/open-uri.rb:224:incatch'
  8: from /cygdrive/c/Data/ruby_test_patch_pthread/lib/open-uri.rb:756:inbuffer_open'
  7: from /cygdrive/c/Data/ruby_test_patch_pthread/lib/open-uri.rb:337:in open_http'
  6: from /cygdrive/c/Data/net/http.rb:922:start'
  5: from /cygdrive/c/Data/net/http.rb:933:indo_start'
  4: from /cygdrive/c/Data/net/http.rb:948:inconnect'
  3: from /cygdrive/c/Data/net/http.rb:948:inblock in timeout'
  2: from /cygdrive/c/Data/net/http.rb:97:inblock in timeout'
  1: from /cygdrive/c/Data/ruby_test_patch_pthread/lib/timeout.rb:97:injoin': Interrupt

Please tell me what I can do to help you solve this problem.

8 - 09/23/2018 08:52 AM - normalperson (Eric Wong)

duerst@it.aoyama.ac.jp wrote:

    Please tell me what I can do to help you solve this problem.

Oops, I guess pthread_kill isn't going to work in cygwin :x

This brings the Cygwin pthreads timer thread closer to what was
in 2.5 by setting ec->interrupt_flag (via rb_threadptr_check_signal):

https://80x24.org/spew/20180923084138.8015-1-x@80x24.org/raw

I suspect the "timer_thread_function()" call is redundant and you
can comment it out, too.

Honestly, if this doesn't work, I don't know how 2.5 and earlier
worked with Cygwin...
Also, does this work if you use other Windows environments which use thread_win32.c instead?

Thanks for your patience.

#9 - 09/25/2018 12:22 AM - duerst (Martin Dürst)

normalperson (Eric Wong) wrote:

This brings the Cygwin pthreads timer thread closer to what was in 2.5 by setting ec->interrupt_flag (via rb_threadptr_check_signal):

https://80x24.org/spew/20180923084138.8015-1-e@80x24.org/raw

Thanks!

Should I use together with or separately from 0001-thread_pthread.c-fall-back-to-UBF_TIMER_PTHREAD-for-.patch ?

#10 - 09/25/2018 12:24 AM - duerst (Martin Dürst)

What I was wondering is: Why do things get stuck in a case where the Timeout isn't actually invoked (as far as I understand). Shouldn't the program just move ahead? Any background information on what may be happening and why is appreciated; it may help me contribute to the solution.

#11 - 09/25/2018 12:42 AM - normalperson (Eric Wong)

duerst@it.aoyama.ac.jp wrote:

normalperson (Eric Wong) wrote:

This brings the Cygwin pthreads timer thread closer to what was in 2.5 by setting ec->interrupt_flag (via rb_threadptr_check_signal):

https://80x24.org/spew/20180923084138.8015-1-e@80x24.org/raw

Thanks!

Should I use together with or separately from 0001-thread_pthread.c-fall-back-to-UBF_TIMER_PTHREAD-for-.patch ?

Together with previous patch.

#12 - 09/25/2018 12:52 AM - normalperson (Eric Wong)

duerst@it.aoyama.ac.jp wrote:

What I was wondering is: Why do things get stuck in a case where the Timeout isn't actually invoked (as far as I understand). Shouldn't the program just move ahead? Any background information on what may be happening and why is appreciated; it may help me contribute to the solution.

I'm actually not sure how things worked at all under 2.5 for Windows with Cygwin; just trying to give you a code path which resembles the 2.5 code paths.

Perhaps usa (Usaku NAKAMURA) or another Windows expert knows how interrupts work?

Is there something which shows system call traces which could be useful in diagnosing this?

/me goes afk again for a while...

#13 - 09/25/2018 06:13 AM - duerst (Martin Dürst)

normalperson (Eric Wong) wrote:

Is there something which shows system call traces which could be useful in diagnosing this?

strace seems to do that, see https://cygwin.com/cygwin-ug-net/strace.html.
strace ./ruby -ropen-uri -e 'open "http://web.mit.edu/index.html"'

produces something like 34 thousand lines out output.

#14 - 09/25/2018 06:32 AM - duerst (Martin Dürst)
I managed to create a very short script that seem so recreate the problem (or at least something similar).

It seems to happen with a Timeout inside a Timeout (I didn't manage to create a problem with only one Timeout).

Here's the script:

```ruby
require 'timeout'

def time_out
  begin
    Timeout.timeout(3) do |_|
      puts 'Before sleeping in method'
      sleep 1
      puts 'After sleeping in method'
    end
    rescue Timeout::Error
      puts 'Timeout triggered in method'
    end
  end
  begin
    Timeout.timeout(5) do |_|
      puts 'Before sleeping'
      sleep 1
      puts 'After sleeping, before method call'
      time_out
      puts 'After method call'
    end
    rescue Timeout::Error
      puts 'Timeout triggered'
  end
end
```

When I run this, I get all the expected output, as follows:

Before sleeping
After sleeping, before method call
Before sleeping in method
After sleeping in method

but this script, rather than stopping after that (2 seconds overall), just hangs (for minutes), until I use Ctrl-C.

At that point, it prints out

```
Traceback (most recent call last):
  ....timeout_test.rb: execution expired (Timeout::Error)
```

#15 - 09/25/2018 09:32 AM - normalperson (Eric Wong)
duerst@it.aoyama.ac.jp wrote:

  normalperson (Eric Wong) wrote:

  Is there something which shows system call traces which could be useful in diagnosing this?

  strace seems to do that, see https://cygwin.com/cygwin-ug-net/strace.html.

OK, same as on Linux, I guess? (`-f` to trace children should be useful.

So, are you also confirming my two patches together do NOT work for you? (I somewhat expected them to, since they try to follow the 2.5 code path...)

strace ./ruby -ropen-uri -e 'open "http://web.mit.edu/index.html"'
Hint #1: --disable-gems to reduce syscalls

Hint #2: add `warn' statements before and after the parts you're interested in, which makes it easier to find the part of the code you want to trace:

e.g.:

    warn "BEGIN FOO"
    .... # whatever you're interested in
    warn "END FOO"

You can the search for "BEGIN FOO"/"END FOO" in the strace output from write/writev syscalls, and everything in between is what you care about.

When I debug in C, I may also use write to a bogus FD to make things easier-to-find:

    write(-13, "blah blah
", 10); errno = 0;

That Cygwin manpage doesn't give strace output examples, so I also wonder if the output is capable of showing strings (I'm spoiled by Linux strace, which does). Sometimes I will use weird/uncommon syscalls or sequences if string matches don't work; but same idea above for finding unique/rare strings.

But yeah, I also have no idea what syscalls you'd see in Windows. At least on Linux I would start looking for where socket() is called along with nearby clone() syscalls for Timeout...

#16 - 09/25/2018 09:32 AM - normalperson (Eric Wong)

duerst@it.aoyama.ac.jp wrote:

    When I run this, I get all the expected output, as follows:
    
    Before sleeping
    After sleeping, before method call
    Before sleeping in method
    After sleeping in method
    
    but this script, rather than stopping after that (2 seconds overall), just hangs (for minutes), until I use Ctrl-C.

If you don't hit Ctrl-C, what is the CPU usage of that script? (from top(1) or similar tools)

Also, that would be a good time to attach "strace -f -p "
and see what is going on. Thanks

#17 - 09/30/2018 09:08 AM - duerst (Martin Dürst)

normalperson (Eric Wong) wrote:

    Eric, many thanks for your answers, and sorry to be late with my replies.
    
    If you don't hit Ctrl-C, what is the CPU usage of that script? (from top(1) or similar tools)

Using the Windows Task Manager, it's 0.0%.

    Also, that would be a good time to attach "strace -f -p "
    and see what is going on. Thanks

Here is what I get:

    $ strace -f -p 21772
    --- Process 14556 created
--- Process 14556 loaded C:\Windows\System32\ntdll.dll at 00007ffaccc0000
--- Process 14556 thread 20116 created
--- Process 14556 thread 14344 created
--- Process 14556 thread 10272 created
--- Process 14556 loaded C:\Windows\System32\kernel32.dll at 00007ffaccb50000
--- Process 14556 loaded C:\Windows\System32\KernelBase.dll at 00007ffaca420000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\bin\cygruby260.dll at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\bin\cygwin1.dll at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\bin\cygcc_s-seh-1.dll at 00000000b5df0000
--- Process 14556 loaded C:\Windows\System32\user32.dll at 00007ffacc910000
--- Process 14556 loaded C:\Windows\System32\gdi32.dll at 00007ffacc690000
--- Process 14556 loaded C:\Windows\System32\advapi32.dll at 00007ffacc1d0000
--- Process 14556 loaded C:\Windows\System32\msvcrt.dll at 00007ffacc700000
--- Process 14556 loaded C:\Windows\System32\sechost.dll at 00007ffacc910000
--- Process 14556 loaded C:\Windows\System32\rpcrt4.dll at 00007ffaca6a0000
--- Process 14556 loaded C:\Windows\System32\sspicli.dll at 00007ffacc9e40000
--- Process 14556 loaded C:\Windows\System32\cryptbase.dll at 00007ffacc8290000
--- Process 14556 loaded C:\Windows\System32\bcryptprimitives.dll at 00007ffacc9e40000
--- Process 14556 loaded C:\Windows\System32\imm32.dll at 00007ffacc5e0000
--- Process 14556 loaded C:\Windows\System32\msctf.dll at 00007ffacc620000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\systems\cygwin64\usr\local\lib\ruby\2.6.0\x86_64-cygwin\enc\trans\transdb.so at 00000000b5d50000
--- Process 14556 loaded C:\windows\system32\ruby.exe
--- Process 14556 thread 14032 created
--- Process 14556 thread 14032 exited with status 0x0
1   1 [sig]  ruby 21772 ****************************************************
84  85 [sig]  ruby 21772 Program name: C:\systems\cygwin64\usr\local\bin\ruby.exe (pid 21772, ppid 632, windows pid 14556)
50  135 [sig]  ruby 21772 OS version: Windows NT-6.3
34  169 [sig]  ruby 21772 ****************************************************

This then also 'hangs', with CPU usage 0.0%. My understanding would be that this means that strace doesn't have much to show because there's nothing going on on the Ruby side.

#18 - 09/30/2018 09:19 AM - duerst (Martin Dürst)
- File strace_after_ctrl_C.txt added

duerst (Martin Dürst) wrote:

This then also 'hangs', with CPU usage 0.0%. My understanding would be that this means that strace doesn't have much to show because there's nothing going on on the Ruby side.

Attached is the rest of the strace output, after I hit Ctrl-C on the Ruby side. Not sure it will help anything.

#19 - 09/30/2018 04:12 PM - MSP-Greg (Greg L)
duerst (Martin Dürst)

I thought that Cygwin & MSYS2/MinGW share some code (or may have some time ago). When running the code above, output as below:

ruby 2.6.0dev (2018-09-30 trunk 64889) [x64-mingw32]
Before sleeping
After sleeping, before method call
Before sleeping in method
After sleeping in method
After method call

FYI, Greg

#20 - 10/10/2018 05:44 AM - duerst (Martin Dürst)
- Subject changed from open-uri hangs on cygwin to Nested Timeouts hang on cygwin

#21 - 10/15/2018 10:42 PM - normalperson (Eric Wong)
duerst@it.aoyama.ac.jp wrote:

File strace_after_ctrl_C.txt added
This then also 'hangs', with CPU usage 0.0%. My understanding would be that this means that strace doesn't have much to show because there's nothing going on on the Ruby side.

Attached is the rest of the strace output, after I hit Ctrl-C on the Ruby side. Not sure it will help anything.

Thanks, but lots of stuff in there I don't understand since it's not Linux...

Did Cygwin use sleepy timer-thread in Ruby 2.5 and earlier?

If not, can you try a 3rd patch to disable sleepy timer-thread?
(thread_win32.c doesn't use sleepy timer thread, either)

This 3rd patch goes on top of the previous two:

https://80x24.org/spew/20181015223141.18303-1-e@80x24.org/raw

The 3 patch series should be:

thread_pthread.c: fall back to UBF_TIMER_PTHREAD for Cygwin
thread_pthread.c: check and reset interrupts for UBF_TIMER_PTHREAD
thread_pthread.c: do not use sleepy timer thread on Cygwin

Footnote: Sleepy timer thread is ideal if your platform supports it because it reduce CPU wakeups and therefore power consumption.

#22 - 10/22/2018 09:03 AM - normalperson (Eric Wong)
Martin: ping on [ruby-core:89410]
http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-core/89410

#23 - 10/23/2018 01:22 AM - normalperson (Eric Wong)

Martin: ping on [ruby-core:89410]

Thanks for the private reply, just wanted to make sure you were aware of it. I barely have time for Ruby in the past month or so 😞

#24 - 11/16/2018 02:43 AM - normalperson (Eric Wong)
Ping again on https://bugs.ruby-lang.org/issues/15130

This 3rd patch goes on top of the previous two:

https://80x24.org/spew/20181015223141.18303-1-e@80x24.org/raw

The 3 patch series should be:

<table>
<thead>
<tr>
<th>thread_pthread.c</th>
<th>Fall back to UBF_TIMER_PTHREAD for Cygwin</th>
</tr>
</thead>
<tbody>
<tr>
<td>thread_pthread.c</td>
<td>Check and reset interrupts for UBF_TIMER_PTHREAD</td>
</tr>
<tr>
<td>thread_pthread.c</td>
<td>Do not use sleepy timer thread on Cygwin</td>
</tr>
</tbody>
</table>

Btw, what is the reason for supporting Cygwin in addition to other Windows builds?

It clutters thread_pthread.c a bit for *nix developers with special cases.

#25 - 11/22/2018 08:17 AM - duerst (Martin Dürst)
- Status changed from Open to Closed

First, I have good news. This error doesn't occur anymore. I discovered this yesterday when I was preparing to apply your third patch, and today I was able to confirm that it also applies without any of the patches you sent, on trunk. I very much hope this error doesn't resurface!

Many thanks for all your help and patience.
normalperson (Eric Wong) wrote:

Btw, what is the reason for supporting Cygwin in addition to other Windows builds?

The simple answer is that same as for mountains: Because it's there.

A slightly longer answer is that for me (and my students), it's quite a good combination between the convenience of Windows and the power of Linux. Because of the current bug, I recently tried to install mingw/msys, but it was too confusing. Maybe it's just because I'm too used to cygwin, but maybe not. I try to avoid Microsoft stuff when I can avoid it, which applies to Visual Studio.

It clutters thread_pthread.c a bit for *nix developers with special cases.

The world would be simpler but more boring if there were only one OS.

#26 - 11/23/2018 04:33 AM - duerst (Martin Dürst)

Just for the record, the good news was at r65920. It may have started quite a bit before that, though.

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001-thread_pthread.c-fall-back-to-UBF_TIMER_PTHREAD-for-.patch</td>
<td>29 KB</td>
<td>09/21/2018</td>
<td>normalperson (Eric Wong)</td>
</tr>
<tr>
<td>strace_after_ctrl_C.txt</td>
<td>11.9 KB</td>
<td>09/30/2018</td>
<td>duerst (Martin Dürst)</td>
</tr>
</tbody>
</table>