

Ruby master - Feature #15002

[PATCH] thread.c (sleep_*): reduce the effect of spurious interrupts

08/17/2018 08:07 AM - normalperson (Eric Wong)

Status:	Closed
Priority:	Normal
Assignee:	normalperson (Eric Wong)
Target version:	
Description	
<p>Spurious interrupts from SIGCHLD cause Mutex#sleep (via ConditionVariable#wait) to return early and breaks some use cases. Since these are outside the programs's control with MJIT, we will only consider pending interrupts (e.g. those from Thread#run) and signals which cause a Ruby-level Signal.trap handler to fire as "spurious" wakeups.</p> <p>This fixes all the Net::IMAP tests in test-all for me with --jit-wait</p> <p>Remaining test-all failures with --jit-wait after this patch:</p> <pre>> 1) Failure: > TestThreadQueue#test_queue_close_multi_multi [/ruby/test/ruby/test_thread_queue.rb:526]: > no threads running</pre> <p>Seems like the test is too timing-sensitive for producer thread lifetime.</p> <pre>> 2) Failure: > TestIO#test_recycled_fd_close [/ruby/test/ruby/test_io.rb:3804]: > Expected /stream closed/ to match "closed stream".</pre> <p>Not sure if solvable with MJIT, the difference is minor and won't affect real-world tests. We should probably loosen the test to allow either "stream closed" or "closed stream"</p> <pre>> 3) Failure: > TestRubyOptimization#test_tailcall_condition_block [/ruby/test/ruby/test_optimization.rb:439]: > [ruby-core:78015] [Bug #12905]: 10079 / 20158 stack levels. > Exception raised: > <#<SystemStackError: stack level too deep>>.</pre> <p>Haven't investigated (I think k0kubun would know better)</p>	

Associated revisions

Revision fe2f89af - 08/18/2018 09:07 AM - normal

thread.c (sleep_*): reduce the effect of spurious interrupts

Spurious interrupts from SIGCHLD cause Mutex#sleep (via ConditionVariable#wait) to return early and breaks some use cases. Since these are outside the programs's control with MJIT, we will only consider pending interrupts (e.g. those from Thread#run) and signals which cause a Ruby-level Signal.trap handler to fire as "spurious" wakeups.

[ruby-core:88537] [Feature #15002]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64444 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 64444 - 08/18/2018 09:07 AM - normalperson (Eric Wong)

thread.c (sleep_*): reduce the effect of spurious interrupts

Spurious interrupts from SIGCHLD cause Mutex#sleep (via ConditionVariable#wait) to return early and breaks some use

cases. Since these are outside the programs's control with MJIT, we will only consider pending interrupts (e.g. those from Thread#run) and signals which cause a Ruby-level Signal.trap handler to fire as "spurious" wakeups.

[ruby-core:88537] [Feature #15002]

Revision 64444 - 08/18/2018 09:07 AM - normal

thread.c (sleep_*): reduce the effect of spurious interrupts

Spurious interrupts from SIGCHLD cause Mutex#sleep (via ConditionVariable#wait) to return early and breaks some use cases. Since these are outside the programs's control with MJIT, we will only consider pending interrupts (e.g. those from Thread#run) and signals which cause a Ruby-level Signal.trap handler to fire as "spurious" wakeups.

[ruby-core:88537] [Feature #15002]

Revision 9e59487a - 08/18/2018 06:29 PM - normal

thread.c (sleep_*): check interrupt before changing th->status

Having threads switch before we sleep can cause applications to misread the state of the thread. Now, we are consistent with blocking_region_begin behavior and change th->status AFTER checking interrupts.

Maybe this can fix [Bug #15002]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64449 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 64449 - 08/18/2018 06:29 PM - normalperson (Eric Wong)

thread.c (sleep_*): check interrupt before changing th->status

Having threads switch before we sleep can cause applications to misread the state of the thread. Now, we are consistent with blocking_region_begin behavior and change th->status AFTER checking interrupts.

Maybe this can fix [Bug #15002]

Revision 64449 - 08/18/2018 06:29 PM - normal

thread.c (sleep_*): check interrupt before changing th->status

Having threads switch before we sleep can cause applications to misread the state of the thread. Now, we are consistent with blocking_region_begin behavior and change th->status AFTER checking interrupts.

Maybe this can fix [Bug #15002]

History

#1 - 08/18/2018 07:51 AM - k0kubun (Takashi Kokubun)

- Assignee changed from k0kubun (Takashi Kokubun) to normalperson (Eric Wong)

Thanks to deal with it. Actually test-all with --jit-wait is running successfully on my Wercker CI and ko1's trunk-mjit-wait for now <http://ci.rvm.jp/results/trunk-mjit-wait@silicon-docker>, but as long as it passes the tests on your environment, it looks good to merge the change to deal with the spurious interrupts by SIGCHLD from MJIT.

Haven't investigated (I think k0kubun would know better)

I haven't investigated yet either.

#2 - 08/18/2018 09:07 AM - normalperson (Eric Wong)

- Status changed from Open to Closed

Applied in changeset [trunk|r64444](#).

thread.c (sleep_*): reduce the effect of spurious interrupts

Spurious interrupts from SIGCHLD cause Mutex#sleep (via ConditionVariable#wait) to return early and breaks some use cases. Since these are outside the programs's control with MJIT, we will only consider pending interrupts (e.g. those from Thread#run) and signals which cause a Ruby-level Signal.trap handler to fire as "spurious" wakeups.

[ruby-core:88537] [Feature [#15002](#)]

#3 - 08/18/2018 09:22 AM - normalperson (Eric Wong)

takashikkbn@gmail.com wrote:

Thanks to deal with it. Actually test-all with --jit-wait is running successfully on my Wercker CI and ko1's trunk-mjit-wait for now <http://ci.rvm.jp/results/trunk-mjit-wait@silicon-docker>, but as long as it passes the tests on your environment, it looks good to merge the change to deal with the spurious interrupts by SIGCHLD from MJIT.

OK, r64444 is committed. It should fix spurious wakeups from the new ConditionVariable#wait specs

TestThreadQueue#test_queue_close_multi_multi [/ruby/test/ruby/test_thread_queue.rb:526]:
no threads running

TestIO#test_recycled_fd_close [/ruby/test/ruby/test_io.rb:3804]:
Expected /stream closed/ to match "closed stream".

So you guys don't see these two failures under CI?
I got them every time...

3) Failure:
TestRubyOptimization#test_tailcall_condition_block [/ruby/test/ruby/test_optimization.rb:439]:
[ruby-core:78015] [Bug [#12905](#)]: 10079 / 20158 stack levels.
Exception raised:
<#>.

This one might be sporadic, I don't think I saw it every time.

Files

0001-thread.c-sleep_-reduce-the-effect-of-spurious-interr.patch	6.17 KB	08/17/2018	normalperson (Eric Wong)
---	---------	------------	--------------------------