

## Backport200 - Backport #7999

### bootstarap test hungs at threading tests

03/02/2013 01:04 PM - ko1 (Koichi Sasada)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	nagachika (Tomoyuki Chikanaga)	
<b>Description</b>		
<p>On [ruby-list:49232], Ootani-san reported that Ruby 2.0.0p0 hungs on bootstarap test hungs at threading tests.</p> <pre>assert_equal 'true', %{\n  t = Thread.new { loop {} }\n  begin\n    pid = fork {\n      exit t.status != "run"\n    }\n    Process.wait pid\n    \$?.success?\n  rescue NotImplementedError\n    true\n  end\n}</pre> <p>He rewrites to " t = Thread.new { loop {sleep 1} }" and it work successfully. His environment is CentOS6.3 and VMware. He doubts this issue caused only on single-core processor because there is no problem on multi-core VM setting.</p> <p>I also observed same issue on VirtualBox with Ubuntu. I haven't check yet that it is only on single-core or not.</p> <p>I want to clarify the reason of this issue. Please give information about your environment if you have same issue.</p>		
<b>Related issues:</b>		
Related to Ruby master - Bug #6098: Re: GVL		<b>Closed</b>

#### Associated revisions

##### Revision bcf12936 - 03/10/2013 04:00 AM - kosaki (Motohiro KOSAKI)

- thread\_pthread.c (ARRAY\_SIZE): new.
- thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39686 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 39686 - 03/10/2013 04:00 AM - kosaki (Motohiro KOSAKI)

- thread\_pthread.c (ARRAY\_SIZE): new.
- thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.

##### Revision 39686 - 03/10/2013 04:00 AM - kosaki (Motohiro KOSAKI)

- thread\_pthread.c (ARRAY\_SIZE): new.
- thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.

##### Revision 39686 - 03/10/2013 04:00 AM - kosaki (Motohiro KOSAKI)

- thread\_pthread.c (ARRAY\_SIZE): new.

- thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.

#### Revision 39686 - 03/10/2013 04:00 AM - kosaki (Motohiro KOSAKI)

- thread\_pthread.c (ARRAY\_SIZE): new.
- thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.

#### Revision 39686 - 03/10/2013 04:00 AM - kosaki (Motohiro KOSAKI)

- thread\_pthread.c (ARRAY\_SIZE): new.
- thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.

#### Revision 39686 - 03/10/2013 04:00 AM - kosaki (Motohiro KOSAKI)

- thread\_pthread.c (ARRAY\_SIZE): new.
- thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.

#### Revision 6901ebcb - 03/11/2013 03:54 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 39679,39682,39683,39685,39686,39694: [Backport #7999]

- \* thread\_pthread.c (set\_nonblock): new helper function for set O\_NONBLOCK.
- \* thread\_pthread.c (rb\_thread\_create\_timer\_thread): set O\_NONBLOCK to timer\_thread\_pipe[0] too.
- \* thread\_pthread.c (consume\_communication\_pipe): retry when read returned CCP\_READ\_BUFF\_SIZE.
- \* thread\_pthread.c (rb\_thread\_create\_timer\_thread): factor out creating communication pipe logic into separate function.
- \* thread\_pthread.c (setup\_communication\_pipe): new helper function.
- \* thread\_pthread.c (set\_nonblock): moves a definition before setup\_communication\_pipe.
- \* thread\_pthread.c (rb\_thread\_wakeup\_timer\_thread\_fd): add fd argument and remove hardcoded dependency of timer\_thread\_pipe[1].
- \* thread\_pthread.c (consume\_communication\_pipe): add fd argument.
- \* thread\_pthread.c (close\_communication\_pipe): ditto.
- \* thread\_pthread.c (timer\_thread\_sleep): adjust the above changes.
- \* thread\_pthread.c (setup\_communication\_pipe\_internal): factor out pipe initialize logic.
- \* thread\_pthread.c (ARRAY\_SIZE): new.
- \* thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- \* thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.
- \* process.c (setup\_communication\_pipe): remove unused function. it was unintentionally added r39683.

**Revision 39727 - 03/11/2013 03:54 PM - nagachika (Tomoyuki Chikanaga)**

merge revision(s) 39679,39682,39683,39685,39686,39694: [Backport #7999]

- \* thread\_pthread.c (set\_nonblock): new helper function for set O\_NONBLOCK.
- \* thread\_pthread.c (rb\_thread\_create\_timer\_thread): set O\_NONBLOCK to timer\_thread\_pipe[0] too.
- \* thread\_pthread.c (consume\_communication\_pipe): retry when read returned CCP\_READ\_BUFF\_SIZE.
- \* thread\_pthread.c (rb\_thread\_create\_timer\_thread): factor out creating communication pipe logic into separate function.
- \* thread\_pthread.c (setup\_communication\_pipe): new helper function.
- \* thread\_pthread.c (set\_nonblock): moves a definition before setup\_communication\_pipe.
- \* thread\_pthread.c (rb\_thread\_wakeup\_timer\_thread\_fd): add fd argument and remove hardcoded dependency of timer\_thread\_pipe[1].
- \* thread\_pthread.c (consume\_communication\_pipe): add fd argument.
- \* thread\_pthread.c (close\_communication\_pipe): ditto.
- \* thread\_pthread.c (timer\_thread\_sleep): adjust the above changes.
- \* thread\_pthread.c (setup\_communication\_pipe\_internal): factor out pipe initialize logic.
- \* thread\_pthread.c (ARRAY\_SIZE): new.
- \* thread\_pthread.c (gvl\_acquire\_common): use low priority notification for avoiding timer thread interval confusion. If we use timer\_thread\_pipe[1], every gvl\_yield() request one more gvl\_yield(). It lead to thread starvation. [Bug #7999] [ruby-core:53095]
- \* thread\_pthread.c (rb\_reserved\_fd\_p): adds timer\_thread\_pipe\_low to reserved fds.
- \* process.c (setup\_communication\_pipe): remove unused function. it was unintentionally added r39683.

## History

---

**#1 - 03/02/2013 01:59 PM - kosaki (Motohiro KOSAKI)**

I could reproduced. try taskset command.

e.g.  
taskset -c 0 ./ruby test.rb

**#2 - 03/02/2013 03:43 PM - kosaki (Motohiro KOSAKI)**

- Status changed from Open to Assigned
- Assignee changed from ko1 (Koichi Sasada) to kosaki (Motohiro KOSAKI)

main thread never exit from rb\_threadptr\_execute\_interrupts() because timer thread wake-up lead to increase t->running\_time\_us and exhaust thread time slice.

**#3 - 03/02/2013 04:01 PM - kosaki (Motohiro KOSAKI)**

This issue was introduced at [r35163](#) (see [Bug#6098]).

**#4 - 03/10/2013 01:00 PM - kosaki (Motohiro KOSAKI)**

- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset [r39686](#).  
Koichi, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

---

- `thread_pthread.c (ARRAY_SIZE)`: new.
- `thread_pthread.c (gvl_acquire_common)`: use low priority notification for avoiding timer thread interval confusion. If we use `timer_thread_pipe[1]`, every `gvl_yield()` request one more `gvl_yield()`. It lead to thread starvation. [Bug [#7999](#)] [ruby-core:53095]
- `thread_pthread.c (rb_reserved_fd_p)`: adds `timer_thread_pipe_low` to reserved fds.

#### #5 - 03/10/2013 01:19 PM - [kosaki \(Motohiro KOSAKI\)](#)

- *Tracker changed from Bug to Backport*
- *Project changed from Ruby master to Backport200*
- *Category deleted (core)*
- *Status changed from Closed to Assigned*
- *Assignee changed from kosaki (Motohiro KOSAKI) to nagachika (Tomoyuki Chikanaga)*
- *Target version deleted (2.1.0)*

Please backport following commits.

[r39679](#) - set `O_NONBLOCK` to `timer_thread_pipe[0]` too.  
[r39682](#) - retry when read returned `CCP_READ_BUFF_SIZE`.  
[r39683](#) - `setup_communication_pipe`: new helper function.  
[r39685](#) - add `fd` argument and remove hardcoded dependency of `timer_thread_pipe`.  
[r39686](#) - use low priority notification for avoiding timer thread interval confusion.

#### #6 - 03/10/2013 03:06 PM - [kosaki \(Motohiro KOSAKI\)](#)

Oops, please backport [r39694](#) too.

#### #7 - 03/12/2013 12:54 AM - [nagachika \(Tomoyuki Chikanaga\)](#)

- *Status changed from Assigned to Closed*

This issue was solved with changeset [r39727](#).  
Koichi, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

---

merge revision(s) [39679,39682,39683,39685,39686,39694](#): [Backport [#7999](#)]

```
* thread_pthread.c (set_nonblock): new helper function for set
O_NONBLOCK.

* thread_pthread.c (rb_thread_create_timer_thread): set O_NONBLOCK
to timer_thread_pipe[0] too.

* thread_pthread.c (consume_communication_pipe): retry when
read returned CCP_READ_BUFF_SIZE.

* thread_pthread.c (rb_thread_create_timer_thread): factor out
creating communication pipe logic into separate function.

* thread_pthread.c (setup_communication_pipe): new helper function.

* thread_pthread.c (set_nonblock): moves a definition before
setup_communication_pipe.

* thread_pthread.c (rb_thread_wakeup_timer_thread_fd): add fd
argument and remove hardcoded dependency of timer_thread_pipe[1].

* thread_pthread.c (consume_communication_pipe): add fd argument.

* thread_pthread.c (close_communication_pipe): ditto.

* thread_pthread.c (timer_thread_sleep): adjust the above changes.

* thread_pthread.c (setup_communication_pipe_internal): factor
```

```
out pipe initialize logic.

* thread_pthread.c (ARRAY_SIZE): new.

* thread_pthread.c (gvl_acquire_common): use low priority
notification for avoiding timer thread interval confusion.
If we use timer_thread_pipe[1], every gvl_yield() request
one more gvl_yield(). It lead to thread starvation.
[Bug #7999] [ruby-core:53095]

* thread_pthread.c (rb_reserved_fd_p): adds timer_thread_pipe_low
to reserved fds.

* process.c (setup_communication_pipe): remove unused function.
it was unintentionally added r39683.
```