Ruby - Bug #5343

Unexpected blocking behavior when interrupt Socket#accept

09/20/2011 02:48 PM - nagachika (Tomoyuki Chikanaga)

Status:	Closed		
Priority:	Normal		
Assignee:	ko1 (Koichi Sasada)		
Target version:	1.9.3		
ruby -v:	-	**Backport:**	

Description

In CentOS release 5.6 (Kernel: 2.6.18-238.12.1.el5, glibc 2.5),
the following sample script rarely (about once every 1000) blocks at Thread#join with 1.9.3-head.

```
require "socket"
require "thread"

queue = Queue.new

th = Thread.start {
s = TCPServer.new(10000)
queue.push(nil)
cli = s.accept
}

queue.pop
th.kill.join
```

Backtrace:

```
thread-1:
#0  0x0000003a3500aee9 in pthread_cond_wait@@GLIBC_2.3.2 ()
from /lib64/libpthread.so.0
#1  0x000000000052d81b in native_cond_wait (th=0x19551550, timeout_tv=0x0)
at ../ruby-1.9.3/thread_pthread.c:307
#2  native_sleep (th=0x19551550, timeout_tv=0x0)
at ../ruby-1.9.3/thread_pthread.c:908
#3  0x000000000052f9ab in sleep_forever (th=0x19551550, deadlockable=1)
at ../ruby-1.9.3/thread.c:855
#4  0x000000000052fa4d in thread_join_sleep (arg=140733838958496)
at ../ruby-1.9.3/thread.c:688
#5  0x0000000000417abb in rb_ensure (b_proc=0x52fa00 <thread_join_sleep>,
data1=140733838958496, e_proc=0x5288e0 <remove_from_join_list>,
data2=140733838958496) at ../ruby-1.9.3/eval.c:736
#6  0x000000000052a78e in thread_join (argc=,
argv=, self=)
at ../ruby-1.9.3/thread.c:721
#7  thread_join_m (argc=, argv=,
self=) at ../ruby-1.9.3/thread.c:802
#8  0x0000000000524b0d in vm_call_cfunc (th=0x19551550, cfp=0x2b38cc37df08,
num=, blockptr=,
flag=, id=, me=0x196663c0,
recv=427644560) at ../ruby-1.9.3/vm_insnhelper.c:404
#9  vm_call_method (th=0x19551550, cfp=0x2b38cc37df08,
num=, blockptr=,
flag=, id=, me=0x196663c0,
recv=427644560) at ../ruby-1.9.3/vm_insnhelper.c:530
#10 0x000000000051908d in vm_exec_core (th=0x19551550,
initial=) at ../ruby-1.9.3/insns.def:1015
#11 0x000000000051ed7e in vm_exec (th=0x19551550) at ../ruby-1.9.3/vm.c:1220
#12 0x0000000000525f9f in rb_iseq_eval_main (iseqval=427473840)
at ../ruby-1.9.3/vm.c:1461
#13 0x0000000000414c22 in ruby_exec_internal (n=0x197abbb0)
at ../ruby-1.9.3/eval.c:204
```

```
#14 0x00000000004172d4 in ruby_exec_node (n=)
at ../ruby-1.9.3/eval.c:251
#15 ruby_run_node (n=) at ../ruby-1.9.3/eval.c:244
#16 0x0000000000414689 in main (argc=2, argv=0x7fff267aa588)
at ../ruby-1.9.3/main.c:38

thread-2:
#0  0x0000003a344cb696 in poll () from /lib64/libc.so.6
#1  0x00000000005301ba in ppoll (fd=,
events=, tv=0x0) at ../ruby-1.9.3/thread.c:2820
#2  rb_wait_for_single_fd (fd=,
events=, tv=0x0) at ../ruby-1.9.3/thread.c:2849
#3  0x000000000053052c in rb_thread_wait_fd_rw (fd=5)
at ../ruby-1.9.3/thread.c:2686
#4  rb_thread_wait_fd (fd=5) at ../ruby-1.9.3/thread.c:2699
#5  0x00002aaaab0b7b6f in rsock_s_accept (klass=427743720, fd=5,
sockaddr=, len=0x40473a3c)
at ../../../ruby-1.9.3/ext/socket/init.c:499
#6  0x00002aaaab0c3310 in tcp_accept (sock=)
at ../../../ruby-1.9.3/ext/socket/tcpserver.c:55
#7  0x0000000000524b0d in vm_call_cfunc (th=0x197ffe90, cfp=0x2aaaab3d3f08,
num=, blockptr=,
flag=, id=, me=0x1980ba70,
recv=427644480) at ../ruby-1.9.3/vm_insnhelper.c:404
#8  vm_call_method (th=0x197ffe90, cfp=0x2aaaab3d3f08,
num=, blockptr=,
flag=, id=, me=0x1980ba70,
recv=427644480) at ../ruby-1.9.3/vm_insnhelper.c:530
#9  0x000000000051908d in vm_exec_core (th=0x197ffe90,
initial=) at ../ruby-1.9.3/insns.def:1015
#10 0x000000000051ed7e in vm_exec (th=0x197ffe90) at ../ruby-1.9.3/vm.c:1220
#11 0x000000000051fad5 in invoke_block_from_c (th=0x197ffe90,
block=, self=, argc=0,
argv=, blockptr=, cref=0x0)
at ../ruby-1.9.3/vm.c:624
#12 0x000000000052026f in rb_vm_invoke_proc (th=0x197ffe90, proc=0x196d72a0,
self=425420560, argc=0, argv=0x197d56c8, blockptr=0x0)
at ../ruby-1.9.3/vm.c:670
#13 0x000000000052f5c1 in thread_start_func_2 (th=0x197ffe90,
stack_start=) at ../ruby-1.9.3/thread.c:453
#14 0x000000000052f75e in thread_start_func_1 (th_ptr=0x197ffe90)
at ../ruby-1.9.3/thread_pthread.c:656
#15 0x0000003a3500673d in start_thread () from /lib64/libpthread.so.0
#16 0x0000003a344d44bd in clone () from /lib64/libc.so.6
```

Associated revisions

Revision f80896c2 - 09/21/2011 05:02 PM - nagachika (Tomoyuki Chikanaga)

- thread_pthread.c (ubf_select): activate timer thread when interrupt
  blocking thread.
  A patch created by Koichi Sasada. [ruby-core:39634] [Bug #5343]
  to cover race condition, timer thread periodically send SIGVTARLM to
  threads in signal thread list. so you should activate timer thread
  when interrupt a thread.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@33307 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 010c8e59 - 09/22/2011 10:45 AM - kosaki (Motohiro KOSAKI)

merge revision(s) 33307:

```
    * thread_pthread.c (ubf_select): activate timer thread when interrupt
      blocking thread.
      A patch created by Koichi Sasada. [ruby-core:39634] [Bug #5343]
      to cover race condition, timer thread periodically send SIGVTARLM to
      threads in signal thread list. so you should activate timer thread
      when interrupt a thread.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@33310 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 8cec0d56 - 12/14/2011 12:23 AM - naruse (Yui NARUSE)

Add test for [Bug #5343] [ruby-core:39634]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34034 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 4c1ab82f0623eca91a95d2a44053be22bbce48ad - 07/29/2018 10:19 PM - Eric Wong

thread_pthread.c (ubf_select): refix [Bug #5343]

We still need to to designate a timer thread after registering target
thread for the ubf list.

Oops :x

Note: I was never able to reproduce
test/thread/test_queue.rb::test_thr_kill failures on my on
Debian machines.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64108 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 4c1ab82f0623eca91a95d2a44053be22bbce48ad - 07/29/2018 10:19 PM - Eric Wong

thread_pthread.c (ubf_select): refix [Bug #5343]

We still need to to designate a timer thread after registering target
thread for the ubf list.

Oops :x

Note: I was never able to reproduce
test/thread/test_queue.rb::test_thr_kill failures on my on
Debian machines.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64108 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 4c1ab82f - 07/29/2018 10:19 PM - Eric Wong

thread_pthread.c (ubf_select): refix [Bug #5343]

We still need to to designate a timer thread after registering target
thread for the ubf list.

Oops :x

Note: I was never able to reproduce
test/thread/test_queue.rb::test_thr_kill failures on my on
Debian machines.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64108 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 26b8a70bb309c7a367b9134045508b5b5a580a77 - 07/30/2018 02:28 AM - Eric Wong

thread_pthread.c (rb_sigwait_sleep): re-fix [Bug #5343] harder

We can't always designate a timer thread, so any sleepers must
also perform ubf wakeups. Note: a similar change needs to be
made for rb_thread_fd_select and rb_wait_for_single_fd.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64111 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 26b8a70bb309c7a367b9134045508b5b5a580a77 - 07/30/2018 02:28 AM - Eric Wong

thread_pthread.c (rb_sigwait_sleep): re-fix [Bug #5343] harder

We can't always designate a timer thread, so any sleepers must
also perform ubf wakeups. Note: a similar change needs to be
made for rb_thread_fd_select and rb_wait_for_single_fd.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64111 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 26b8a70b - 07/30/2018 02:28 AM - Eric Wong

thread_pthread.c (rb_sigwait_sleep): re-fix [Bug #5343] harder

We can't always designate a timer thread, so any sleepers must
also perform ubf wakeups. Note: a similar change needs to be
made for rb_thread_fd_select and rb_wait_for_single_fd.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64111 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 4514362948fdb914c6138b12d961d92e9c0fee6c - 07/30/2018 04:28 AM - Eric Wong

thread_pthread.c (rb_sigwait_sleep): fix uninitialized poll set in UBF case

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64113 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 4514362948fdb914c6138b12d961d92e9c0fee6c - 07/30/2018 04:28 AM - Eric Wong

thread_pthread.c (rb_sigwait_sleep): fix uninitialized poll set in UBF case

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64113 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45143629 - 07/30/2018 04:28 AM - Eric Wong

thread_pthread.c (rb_sigwait_sleep): fix uninitialized poll set in UBF case

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64113 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision ab47a57a46e70634d049e4da20a5441c7a14cdec - 07/30/2018 05:51 AM - Eric Wong

thread*.c: waiting on sigwait_fd performs periodic ubf wakeups

We need to be able to perform periodic ubf_list wakeups when a
thread is sleeping and waiting on signals.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64115 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision ab47a57a46e70634d049e4da20a5441c7a14cdec - 07/30/2018 05:51 AM - Eric Wong

thread*.c: waiting on sigwait_fd performs periodic ubf wakeups

We need to be able to perform periodic ubf_list wakeups when a
thread is sleeping and waiting on signals.

[ruby-core:88088] [Misc #14937] [Bug #5343]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64115 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision ab47a57a - 07/30/2018 05:51 AM - Eric Wong

thread*.c: waiting on sigwait_fd performs periodic ubf wakeups

We need to be able to perform periodic ubf_list wakeups when a
thread is sleeping and waiting on signals.

[ruby-core:88088] [Misc #14937] [Bug #5343] git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64115 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 194a6a2c68e9c8a3536b24db18ceac87535a6051 - 08/06/2018 05:22 AM - Eric Wong

thread_pthread.c: restore timer-thread for now :<

[ruby-core:88306]

Revert "process.c: ensure th->interrupt lock is held when migrating"

This reverts commit 5ca416bdf6b6785cb20f139c2c514eda005fe42f (r64201)

Revert "process.c (rb_waitpid): reduce sigwait_fd bouncing"

This reverts commit 217bdd776fbeea3bfd0b9324eefbfcec3b1ccb3e (r64200).

Revert "test/ruby/test_thread.rb (test_thread_timer_and_interrupt): add timeouts"

This reverts commit 9f395f11202fc3c7edbd76f5aa6ce1f8a1e752a9 (r64199).

Revert "thread_pthread.c (native_sleep): reduce ppoll sleeps"

This reverts commit b3aa256c4d43d3d7e9975ec18eb127f45f623c9b (r64193).

Revert "thread.c (consume_communication_pipe): do not retry after short read"

This reverts commit 291a82f748de56e65fac10edefc51ec7a54a82d4 (r64185).

Revert "test/ruby/test_io.rb (test_race_gets_and_close): timeout each thread"

This reverts commit 3dbd8d1f66537f968f0461ed8547460b3b1241b3 (r64184).

Revert "thread_pthread.c (gvl_acquire_common): persist timeout across calls"

This reverts commit 8c2ae6e3ed072b06fc3cbc34fa8a14b2acbb49d5 (r64165).

Revert "test/ruby/test_io.rb (test_race_gets_and_close): use SIGABRT on timeout"

This reverts commit 931cda4db8afd6b544a8d85a6815765a9c417213 (r64135).

Revert "thread_pthread.c (gvl_yield): do ubf wakeups when uncontended"

This reverts commit 508f00314f46c08b6e9b0141c01355d24954260c (r64133).

Revert "thread_pthread.h (native_thread_data): split condvars on some platforms"

This reverts commit a038bf238bd9a24bf1e1622f618a27db261fc91b (r64124).

Revert "process.c (waitpid_nogvl): prevent conflicting use of sleep_cond"

This reverts commit 7018acc946882f21d519af7c42ccf84b22a46b27 (r64117).

Revert "thread_pthread.c (rb_sigwait_sleep): th may be 0 from MJIT"

This reverts commit 56491afc7916fb24f5c4dc2c632fb93fa7063992 (r64116).

Revert "thread*.c: waiting on sigwait_fd performs periodic ubf wakeups"

This reverts commit ab47a57a46e70634d049e4da20a5441c7a14cdec (r64115).

Revert "thread_pthread.c (gvl_destroy): make no-op on GVL bits"

This reverts commit 95cae748171f4754b97f4ba54da2ae62a8d484fd (r64114).

Revert "thread_pthread.c (rb_sigwait_sleep): fix uninitialized poll set in UBF case"

This reverts commit 4514362948fdb914c6138b12d961d92e9c0fee6c (r64113).

Revert "thread_pthread.c (rb_sigwait_sleep): re-fix [Bug #5343] harder"

This reverts commit 26b8a70bb309c7a367b9134045508b5b5a580a77 (r64111).

Revert "thread.c: move ppoll wrapper into thread_pthread.c"

This reverts commit 3dc7727d22fecbc355597edda25d2a245bf55ba1 (r64110).

Revert "thread.c: move ppoll wrapper before thread_pthread.c"

This reverts commit 2fa1e2e3c3c5c4b3ce84730dee4bcbe9d81b8e35 (r64109).

Revert "thread_pthread.c (ubf_select): refix [Bug #5343]"

This reverts commit 4c1ab82f0623eca91a95d2a44053be22bbce48ad (r64108).

Revert "thread_win32.c: suppress warnings by -Wsuggest-attribute"

This reverts commit 6a9b63e39075c53870933fbac5c1065f7d22047c (r64159).

Revert "thread_pthread: remove timer-thread by restructuring GVL"

This reverts commit 708bfd21156828526fe72de2cedecfaca6647dc1 (r64107).

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64203 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 194a6a2c68e9c8a3536b24db18ceac87535a6051 - 08/06/2018 05:22 AM - Eric Wong

thread_pthread.c: restore timer-thread for now :<

[ruby-core:88306]

Revert "process.c: ensure th->interrupt lock is held when migrating"

This reverts commit 5ca416bdf6b6785cb20f139c2c514eda005fe42f (r64201)

Revert "process.c (rb_waitpid): reduce sigwait_fd bouncing"

This reverts commit 217bdd776fbeea3bfd0b9324eefbfcec3b1ccb3e (r64200).

Revert "test/ruby/test_thread.rb (test_thread_timer_and_interrupt): add timeouts"

This reverts commit 9f395f11202fc3c7edbd76f5aa6ce1f8a1e752a9 (r64199).

Revert "thread_pthread.c (native_sleep): reduce ppoll sleeps"

This reverts commit b3aa256c4d43d3d7e9975ec18eb127f45f623c9b (r64193).

Revert "thread.c (consume_communication_pipe): do not retry after short read"

This reverts commit 291a82f748de56e65fac10edefc51ec7a54a82d4 (r64185).

Revert "test/ruby/test_io.rb (test_race_gets_and_close): timeout each thread"

This reverts commit 3dbd8d1f66537f968f0461ed8547460b3b1241b3 (r64184).

Revert "thread_pthread.c (gvl_acquire_common): persist timeout across calls"

This reverts commit 8c2ae6e3ed072b06fc3cbc34fa8a14b2acbb49d5 (r64165).

Revert "test/ruby/test_io.rb (test_race_gets_and_close): use SIGABRT on timeout"

This reverts commit 931cda4db8afd6b544a8d85a6815765a9c417213 (r64135).

Revert "thread_pthread.c (gvl_yield): do ubf wakeups when uncontended"

This reverts commit 508f00314f46c08b6e9b0141c01355d24954260c (r64133).

Revert "thread_pthread.h (native_thread_data): split condvars on some platforms"

This reverts commit a038bf238bd9a24bf1e1622f618a27db261fc91b (r64124).

Revert "process.c (waitpid_nogvl): prevent conflicting use of sleep_cond"

This reverts commit 7018acc946882f21d519af7c42ccf84b22a46b27 (r64117).

Revert "thread_pthread.c (rb_sigwait_sleep): th may be 0 from MJIT"

This reverts commit 56491afc7916fb24f5c4dc2c632fb93fa7063992 (r64116).

Revert "thread*.c: waiting on sigwait_fd performs periodic ubf wakeups"

This reverts commit ab47a57a46e70634d049e4da20a5441c7a14cdec (r64115).

Revert "thread_pthread.c (gvl_destroy): make no-op on GVL bits"

This reverts commit 95cae748171f4754b97f4ba54da2ae62a8d484fd (r64114).

Revert "thread_pthread.c (rb_sigwait_sleep): fix uninitialized poll set in UBF case"

This reverts commit 4514362948fdb914c6138b12d961d92e9c0fee6c (r64113).

Revert "thread_pthread.c (rb_sigwait_sleep): re-fix [Bug #5343] harder"

This reverts commit 26b8a70bb309c7a367b9134045508b5b5a580a77 (r64111).

Revert "thread.c: move ppoll wrapper into thread_pthread.c"

This reverts commit 3dc7727d22fecbc355597edda25d2a245bf55ba1 (r64110).

Revert "thread.c: move ppoll wrapper before thread_pthread.c"

This reverts commit 2fa1e2e3c3c5c4b3ce84730dee4bcbe9d81b8e35 (r64109).

Revert "thread_pthread.c (ubf_select): refix [Bug #5343]"

This reverts commit 4c1ab82f0623eca91a95d2a44053be22bbce48ad (r64108).

Revert "thread_win32.c: suppress warnings by -Wsuggest-attribute"

This reverts commit 6a9b63e39075c53870933fbac5c1065f7d22047c (r64159).

Revert "thread_pthread: remove timer-thread by restructuring GVL"

This reverts commit 708bfd21156828526fe72de2cedecfaca6647dc1 (r64107).

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64203 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 194a6a2c - 08/06/2018 05:22 AM - Eric Wong

thread_pthread.c: restore timer-thread for now :<

[ruby-core:88306]

Revert "process.c: ensure th->interrupt lock is held when migrating"

This reverts commit 5ca416bdf6b6785cb20f139c2c514eda005fe42f (r64201)

Revert "process.c (rb_waitpid): reduce sigwait_fd bouncing"

This reverts commit 217bdd776fbeea3bfd0b9324eefbfcec3b1ccb3e (r64200).

Revert "test/ruby/test_thread.rb (test_thread_timer_and_interrupt): add timeouts"

This reverts commit 9f395f11202fc3c7edbd76f5aa6ce1f8a1e752a9 (r64199).

Revert "thread_pthread.c (native_sleep): reduce ppoll sleeps"

This reverts commit b3aa256c4d43d3d7e9975ec18eb127f45f623c9b (r64193).

Revert "thread.c (consume_communication_pipe): do not retry after short read"

This reverts commit 291a82f748de56e65fac10edefc51ec7a54a82d4 (r64185).

Revert "test/ruby/test_io.rb (test_race_gets_and_close): timeout each thread"

This reverts commit 3dbd8d1f66537f968f0461ed8547460b3b1241b3 (r64184).

Revert "thread_pthread.c (gvl_acquire_common): persist timeout across calls"

This reverts commit 8c2ae6e3ed072b06fc3cbc34fa8a14b2acbb49d5 (r64165).

Revert "test/ruby/test_io.rb (test_race_gets_and_close): use SIGABRT on timeout"

This reverts commit 931cda4db8afd6b544a8d85a6815765a9c417213 (r64135).

Revert "thread_pthread.c (gvl_yield): do ubf wakeups when uncontended"

This reverts commit 508f00314f46c08b6e9b0141c01355d24954260c (r64133).

Revert "thread_pthread.h (native_thread_data): split condvars on some platforms"

This reverts commit a038bf238bd9a24bf1e1622f618a27db261fc91b (r64124).

Revert "process.c (waitpid_nogvl): prevent conflicting use of sleep_cond"

This reverts commit 7018acc946882f21d519af7c42ccf84b22a46b27 (r64117).

Revert "thread_pthread.c (rb_sigwait_sleep): th may be 0 from MJIT"

This reverts commit 56491afc7916fb24f5c4dc2c632fb93fa7063992 (r64116).

Revert "thread*.c: waiting on sigwait_fd performs periodic ubf wakeups"

This reverts commit ab47a57a46e70634d049e4da20a5441c7a14cdec (r64115).

Revert "thread_pthread.c (gvl_destroy): make no-op on GVL bits"

This reverts commit 95cae748171f4754b97f4ba54da2ae62a8d484fd (r64114).

Revert "thread_pthread.c (rb_sigwait_sleep): fix uninitialized poll set in UBF case"

This reverts commit 4514362948fdb914c6138b12d961d92e9c0fee6c (r64113).

Revert "thread_pthread.c (rb_sigwait_sleep): re-fix [Bug #5343] harder"

This reverts commit 26b8a70bb309c7a367b9134045508b5b5a580a77 (r64111).

Revert "thread.c: move ppoll wrapper into thread_pthread.c"

This reverts commit 3dc7727d22fecbc355597edda25d2a245bf55ba1 (r64110).

Revert "thread.c: move ppoll wrapper before thread_pthread.c"

This reverts commit 2fa1e2e3c3c5c4b3ce84730dee4bcbe9d81b8e35 (r64109).

Revert "thread_pthread.c (ubf_select): refix [Bug #5343]"

This reverts commit 4c1ab82f0623eca91a95d2a44053be22bbce48ad (r64108).

Revert "thread_win32.c: suppress warnings by -Wsuggest-attribute"

This reverts commit 6a9b63e39075c53870933fbac5c1065f7d22047c (r64159).

Revert "thread_pthread: remove timer-thread by restructuring GVL"

This reverts commit 708bfd21156828526fe72de2cedecfaca6647dc1 (r64107).

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@64203 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 09/20/2011 06:22 PM - nagachika (Tomoyuki Chikanaga)

Hi,

I've found that this issue is not specific problem of TCPServer#accept.

The following script also rarely blocks forever.

```
require "thread"

queue = Queue.new
r, w = IO.pipe
th = Thread.start {
queue.push(nil)
r.read 1
}
queue.pop
th.kill.join
```

I guess this is because SIGVTALRM from send from ubf_select() is received by blocking thread after GVL release but before enter poll(2)/ppoll(2).

#2 - 09/20/2011 07:14 PM - kosaki (Motohiro KOSAKI)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

- Priority changed from Normal to 5

Aghh. This is regression since 1.9.3.

#3 - 09/20/2011 11:53 PM - ko1 (Koichi Sasada)

- ruby -v changed from ruby 1.9.3dev (2011-09-17 revision 33290) [i686-linux] to -

(2011/09/20 2:22), Tomoyuki Chikanaga wrote:

> Issue #5343 has been updated by Tomoyuki Chikanaga.
>
> Hi,
>
> I've found that this issue is not specific problem of TCPServer#accept.
>
> The following script also rarely blocks forever.
>
> require "thread"
>
> queue = Queue.new
> r, w = IO.pipe
> th = Thread.start {
> queue.push(nil)
> r.read 1
> }
> queue.pop
> th.kill.join
>
> I guess this is because SIGVTALRM from send from ubf_select() is received by blocking thread after GVL release but before enter poll(2)/ppoll(2).

Thanks. I understand. How about it?

Index: thread_pthread.c

--- thread_pthread.c (revision 33272)
+++ thread_pthread.c (working copy)
@@ -1013,6 +1013,7 @@
{
rb_thread_t *th = (rb_thread_t *)ptr;
add_signal_thread_list(th);

- rb_thread_wakeup_timer_thread(); /* activate timer thread */
  ubf_select_each(th);
  }

--
// SASADA Koichi at atdot dot net

#4 - 09/21/2011 05:59 AM - normalperson (Eric Wong)

SASADA Koichi ko1@atdot.net wrote:

> (2011/09/20 2:22), Tomoyuki Chikanaga wrote:
>
> > I guess this is because SIGVTALRM from send from ubf_select() is received by blocking thread after GVL release but before enter poll(2)/ppoll(2).
>
> Thanks. I understand. How about it?

I'm not sure issues like this are completely avoidable, but your patch
seems to help avoid it.

I needed ~15K tries to reproduce blocking on the original test case
(no SMP on this Xen VM). So far I'm at 65K loop interations and still
going... I will update if I notice it blocked again.

#5 - 09/21/2011 09:25 AM - nagachika (Tomoyuki Chikanaga)

Hi, sasada san.
With your patch, over 60000 times trials pass successfully.
Your patch works fine for me too.
Thank you.

#6 - 09/21/2011 09:53 AM - ko1 (Koichi Sasada)

Thanks Eric and Chikanaga-san.

Could you commit it? Maybe 1.9.3 also needs this patch. But I'm not
sure how to commit frozen 1.9.3 branch.

Regards,
Koichi

(2011/09/20 17:25), Tomoyuki Chikanaga wrote:

> Issue #5343 has been updated by Tomoyuki Chikanaga.
>
> # Hi, sasada san.
With your patch, over 60000 times trials pass successfully.
Your patch works fine for me too.
Thank you.
>
> Bug #5343: Unexpected blocking behavior when interrupt Socket#accept
> http://redmine.ruby-lang.org/issues/5343
>
> Author: Tomoyuki Chikanaga
> Status: Assigned
> Priority: High
> Assignee: Koichi Sasada
> Category: ext
> Target version: 1.9.3
> ruby -v: -
>
> In CentOS release 5.6 (Kernel: 2.6.18-238.12.1.el5, glibc 2.5),
> the following sample script rarely (about once every 1000) blocks at Thread#join with 1.9.3-head.
>
> ```
> require "socket"
> require "thread"
>
> queue = Queue.new
>
> th = Thread.start {
> s = TCPServer.new(10000)
> queue.push(nil)
> cli = s.accept
> }
>
> queue.pop
> th.kill.join
> ```
>
> Backtrace:
>
> ```
> thread-1:
> #0  0x0000003a3500aee9 in pthread_cond_wait@@GLIBC_2.3.2 ()
> from /lib64/libpthread.so.0
> #1  0x000000000052d81b in native_cond_wait (th=0x19551550, timeout_tv=0x0)
> at ../ruby-1.9.3/thread_pthread.c:307
> #2  native_sleep (th=0x19551550, timeout_tv=0x0)
> at ../ruby-1.9.3/thread_pthread.c:908
> #3  0x000000000052f9ab in sleep_forever (th=0x19551550, deadlockable=1)
> at ../ruby-1.9.3/thread.c:855
> #4  0x000000000052fa4d in thread_join_sleep (arg=140733838958496)
> at ../ruby-1.9.3/thread.c:688
> #5  0x0000000000417abb in rb_ensure (b_proc=0x52fa00 <thread_join_sleep>,
> data1=140733838958496, e_proc=0x5288e0 <remove_from_join_list>,
> data2=140733838958496) at ../ruby-1.9.3/eval.c:736
> #6  0x000000000052a78e in thread_join (argc=,
> argv=, self=)
> at ../ruby-1.9.3/thread.c:721
> #7  thread_join_m (argc=, argv=,
> ```

```
self=) at ../ruby-1.9.3/thread.c:802
#8  0x0000000000524b0d in vm_call_cfunc (th=0x19551550, cfp=0x2b38cc37df08,
num=, blockptr=,
flag=, id=, me=0x196663c0,
recv=427644560) at ../ruby-1.9.3/vm_insnhelper.c:404
#9  vm_call_method (th=0x19551550, cfp=0x2b38cc37df08,
num=, blockptr=,
flag=, id=, me=0x196663c0,
recv=427644560) at ../ruby-1.9.3/vm_insnhelper.c:530
#10 0x000000000051908d in vm_exec_core (th=0x19551550,
initial=) at ../ruby-1.9.3/insns.def:1015
#11 0x000000000051ed7e in vm_exec (th=0x19551550) at ../ruby-1.9.3/vm.c:1220
#12 0x0000000000525f9f in rb_iseq_eval_main (iseqval=427473840)
at ../ruby-1.9.3/vm.c:1461
#13 0x0000000000414c22 in ruby_exec_internal (n=0x197abbb0)
at ../ruby-1.9.3/eval.c:204
#14 0x00000000004172d4 in ruby_exec_node (n=)
at ../ruby-1.9.3/eval.c:251
#15 ruby_run_node (n=) at ../ruby-1.9.3/eval.c:244
#16 0x0000000000414689 in main (argc=2, argv=0x7fff267aa588)
at ../ruby-1.9.3/main.c:38

thread-2:
#0  0x0000003a344cb696 in poll () from /lib64/libc.so.6
#1  0x00000000005301ba in ppoll (fd=,
events=, tv=0x0) at ../ruby-1.9.3/thread.c:2820
#2  rb_wait_for_single_fd (fd=,
events=, tv=0x0) at ../ruby-1.9.3/thread.c:2849
#3  0x000000000053052c in rb_thread_wait_fd_rw (fd=5)
at ../ruby-1.9.3/thread.c:2686
#4  rb_thread_wait_fd (fd=5) at ../ruby-1.9.3/thread.c:2699
#5  0x00002aaaab0b7b6f in rsock_s_accept (klass=427743720, fd=5,
sockaddr=, len=0x40473a3c)
at ../../../ruby-1.9.3/ext/socket/init.c:499
#6  0x00002aaaab0c3310 in tcp_accept (sock=)
at ../../../ruby-1.9.3/ext/socket/tcpserver.c:55
#7  0x0000000000524b0d in vm_call_cfunc (th=0x197ffe90, cfp=0x2aaaab3d3f08,
num=, blockptr=,
flag=, id=, me=0x1980ba70,
recv=427644480) at ../ruby-1.9.3/vm_insnhelper.c:404
#8  vm_call_method (th=0x197ffe90, cfp=0x2aaaab3d3f08,
num=, blockptr=,
flag=, id=, me=0x1980ba70,
recv=427644480) at ../ruby-1.9.3/vm_insnhelper.c:530
#9  0x000000000051908d in vm_exec_core (th=0x197ffe90,
initial=) at ../ruby-1.9.3/insns.def:1015
#10 0x000000000051ed7e in vm_exec (th=0x197ffe90) at ../ruby-1.9.3/vm.c:1220
#11 0x000000000051fad5 in invoke_block_from_c (th=0x197ffe90,
block=, self=, argc=0,
argv=, blockptr=, cref=0x0)
at ../ruby-1.9.3/vm.c:624
#12 0x000000000052026f in rb_vm_invoke_proc (th=0x197ffe90, proc=0x196d72a0,
self=425420560, argc=0, argv=0x197d56c8, blockptr=0x0)
at ../ruby-1.9.3/vm.c:670
#13 0x000000000052f5c1 in thread_start_func_2 (th=0x197ffe90,
stack_start=) at ../ruby-1.9.3/thread.c:453
#14 0x000000000052f75e in thread_start_func_1 (th_ptr=0x197ffe90)
at ../ruby-1.9.3/thread_pthread.c:656
#15 0x0000003a3500673d in start_thread () from /lib64/libpthread.so.0
#16 0x0000003a344d44bd in clone () from /lib64/libc.so.6
```

--
// SASADA Koichi at atdot dot net

#7 - 09/22/2011 01:38 AM - nagachika (Tomoyuki Chikanaga)

Koichi Sasada wrote:

> Could you commit it? Maybe 1.9.3 also needs this patch. But I'm not
> sure how to commit frozen 1.9.3 branch.
> I'd be happy to commit it. I'll check in it to trunk and create a backport request ticket.

BTW, let me summarize the reason why this patch works fine. If I misunderstand, please point out.

When interrupt other thread (cf. by Thread#kill, Thread#raise) in blocking region, unblocking function (ubf) is called to interrupt blocking system call like select/ppoll etc... The default ubf (ubf_select) call pthread_kill() to send SIGVTALRM to interrupted thread.
But There's race condition. If interrupted thread was just released GVL and received SIGVTALRM before entering blocking systemcall, signal was lost.
To cover this race condition, ubf_select() also call add_signal_thread_list() to register "signal thread list". Timer thread periodically send SIGVTALRM to threads in signal thread list. Recently timer thread is suspended when there's no more than one runnable thread. So when interrupt other thread and register to signal thread list, you should activate timer thread to polling-mode.

#8 - 09/22/2011 02:02 AM - nagachika (Tomoyuki Chikanaga)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r33307.
Tomoyuki, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- thread_pthread.c (ubf_select): activate timer thread when interrupt
  blocking thread.
  A patch created by Koichi Sasada. [ruby-core:39634] [Bug #5343]
  to cover race condition, timer thread periodically send SIGVTARLM to
  threads in signal thread list. so you should activate timer thread
  when interrupt a thread.

#9 - 09/22/2011 02:53 AM - ko1 (Koichi Sasada)

(2011/09/21 9:38), Tomoyuki Chikanaga wrote:

> > Could you commit it? Maybe 1.9.3 also needs this patch. But I'm not
> > sure how to commit frozen 1.9.3 branch.
> > I'd be happy to commit it. I'll check in it to trunk and create a backport request ticket.

Thanks.

> BTW, let me summarize the reason why this patch works fine. If I misunderstand, please point out.
> When interrupt other thread (cf. by Thread#kill, Thread#raise) in blocking region, unblocking function (ubf) is called to interrupt blocking system call like select/ppoll etc...
> The default ubf (ubf_select)

No. ubf_select is not a default one. ubf_select is only for canceling
select() blocking. If you use it for other process, it may be a bug.

> call pthread_kill() to send SIGVTALRM to interrupted thread.
> But There's race condition.
> If interrupted thread was just released GVL and received SIGVTALRM
> before entering blocking systemcall, signal was lost.
> To cover this race condition, ubf_select() also call
> add_signal_thread_list() to register "signal thread list".
> Timer thread periodically send SIGVTALRM to threads in signal thread
> list.
> Recently timer thread is suspended when there's no more than one
> runnable thread. So when interrupt other thread and register to
> signal thread list, you should activate timer thread to
> polling-mode.

Yes, absolutely.

--
// SASADA Koichi at atdot dot net

#10 - 09/22/2011 04:23 AM - normalperson (Eric Wong)

Tomoyuki Chikanaga nagachika00@gmail.com wrote:

> To cover this race condition, ubf_select() also call
> add_signal_thread_list() to register "signal thread list". Timer
> thread periodically send SIGVTALRM to threads in signal thread list.

Ah, I missed the second part of that earlier. My concern raised in
[ruby-core:39643] is no more, thank you.