

Ruby - Bug #2339

DRb leaves connections half-open

11/06/2009 02:44 AM - pierre@mouraf.org (Pierre-Alexandre Meyer)

Status:	Closed	
Priority:	Normal	
Assignee:	seki (Masatoshi Seki)	
Target version:		
ruby -v:	ruby 1.9.1p243 (2009-07-16 revision 24175) [x86_64-linux]	Backport: 2.5: UNKNOWN, 2.6: UNKNOWN

Description

=begin
Hello,

While examining some tcp frames, I encountered the following behaviour in the DRb protocol.

In a nutshell, when the client closes the connection with a server, a stack trace is sent back to the client (indicating a DRbConnError) leaving the connection half-open. I was wondering if there is a rational behind it?

With the simple server/client example from the DRb documentation (attached), 1707 bytes are being transmitted as part of the tcp conversation, 730 of which is the data for this stack trace.

The behaviour seems to come from drb/drbrb.rb, in DRbServer#main_loop:

```
client.send_reply(succ, result) rescue nil
```

the server sends data back, even when the client tries to close the connection (after initiating a FIN).

The current TCP conversation looks like:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	::1	::1	TCP	49630 > msgsrvr [SYN] Seq=0 Win=32752 Len=0 MSS=16376 TSV=111955923 TSER=0 WS=7
2	0.000014	::1	::1	TCP	msgsrvr > 49630 [SYN, ACK] Seq=0 Ack=1 Win=32728 Len=0 MSS=16376 TSV=111955923 TSER=111955923 WS=7
3	0.000025	::1	::1	TCP	49630 > msgsrvr [ACK] Seq=1 Ack=1 Win=32768 Len=0 TSV=111955923 TSER=111955923
4	0.005520	::1	::1	TCP	49630 > msgsrvr [PSH, ACK] Seq=1 Ack=1 Win=32768 Len=68 TSV=111955923 TSER=111955923
5	0.005533	::1	::1	TCP	msgsrvr > 49630 [ACK] Seq=1 Ack=69 Win=32768 Len=0 TSV=111955923 TSER=111955923
6	0.005947	::1	::1	TCP	msgsrvr > 49630 [PSH, ACK] Seq=1 Ack=69 Win=32768 Len=45 TSV=111955923 TSER=111955923
7	0.005956	::1	::1	TCP	49630 > msgsrvr [ACK] Seq=69 Ack=46 Win=32768 Len=0 TSV=111955923 TSER=111955923
8	0.008436	::1	::1	TCP	49630 > msgsrvr [FIN, ACK] Seq=69 Ack=46 Win=32768 Len=0 TSV=111955924 TSER=111955923
9	0.009124	::1	::1	TCP	msgsrvr > 49630 [PSH, ACK] Seq=46 Ack=70 Win=32768 Len=730 TSV=111955924 TSER=111955924
10	0.009142	::1	::1	TCP	49630 > msgsrvr [RST] Seq=70 Win=0 Len=0

The extra PUSH is

```
0000 00 00 00 00 00 00 00 00 00 00 00 00 86 dd 60 00 ..... 0010 00 00 02 fa 06 40 00 00 00 00 00 00 00 00 00 00
.....@..... 0020 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 ..... 0030 00 00 00 00 00 01 22 53 c1 de ae c9 dc 23
ae b0 ..... "S.....#.. 0040 48 3c 80 18 01 00 03 02 00 00 01 01 08 0a 06 ac H<..... 0050 4f d4 06 ac 4f d4 00 00 00 03 04 08
46 00 00 02 O...O.....F... 0060 cf 04 08 6f 3a 16 44 52 62 3a 3a 44 52 62 43 6f ...o:..DRb::DRbCo 0070 6e 6e 45 72 72 6f 72 07
3a 09 6d 65 73 67 49 22 nnError:..mesgl" 0080 16 63 6f 6e 6e 65 63 74 69 6f 6e 20 63 6c 6f 73 ..connection clos 0090 65 64 06
3a 0d 65 6e 63 6f 64 69 6e 67 22 0d 55 ed:..encoding".U 00a0 53 2d 41 53 43 49 49 3a 07 62 74 5b 0e 22 41 2f S-ASCII:..bt[".A/
00b0 6f 70 74 2f 72 75 62 79 2d 31 2e 39 2e 31 2d 70 opt/ruby-1.9.1-p 00c0 32 34 33 2f 6c 69 62 2f 72 75 62 79 2f 31 2e 39
```

243/lib/ruby/1.9.0/0d0 2e 31 2f 64 72 62 2f 64 72 62 2e 72 62 3a 35 37 .1/drbrb:57 00e0 30 3a 69 6e 20 60 6c 6f 61 64 27 22 49 2f 6f 70 0:in load""/op
00f0 74 2f 72 75 62 79 2d 31 2e 39 2e 31 2d 70 32 34 t/ruby-1.9.1-p24
0100 33 2f 6c 69 62 2f 72 75 62 79 2f 31 2e 39 2e 31 3/lib/ruby/1.9.1
0110 2f 64 72 62 2f 64 72 62 2e 72 62 3a 36 31 32 3a /drbrb:612:
0120 69 6e 20 60 72 65 63 76 5f 72 65 71 75 65 73 74 in recv_request 0130 27 22 49 2f 6f 70 74 2f 72 75 62 79 2d 31 2e 39
""/opt/ruby-1.9.0/140 2e 31 2d 70 32 34 33 2f 6c 69 62 2f 72 75 62 79 .1-p243/lib/ruby 0150 2f 31 2e 39 2e 31 2f 64 72 62 2f 64 72 62 2e 72 /1.9.1/drbrb:0160 62 3a 39 30 38 3a 69 6e 20 60 72 65 63 76 5f 72 b:908:in recv_r
0170 65 71 75 65 73 74 27 22 4e 2f 6f 70 74 2f 72 75 equest""N/opt/ru
0180 62 79 2d 31 2e 39 2e 31 2d 70 32 34 33 2f 6c 69 by-1.9.1-p243/li
0190 62 2f 72 75 62 79 2f 31 2e 39 2e 31 2f 64 72 62 b/ruby/1.9.1/drbr
01a0 2f 64 72 62 2e 72 62 3a 31 35 33 33 3a 69 6e 20 /drbrb:1533:in
01b0 60 69 6e 69 74 5f 77 69 74 68 5f 63 6c 69 65 6e init_with_clien 01c0 74 27 22 4b 2f 6f 70 74 2f 72 75 62 79 2d 31 2e
t""K/opt/ruby-1.01d0 39 2e 31 2d 70 32 34 33 2f 6c 69 62 2f 72 75 62 9.1-p243/lib/rub 01e0 79 2f 31 2e 39 2e 31 2f 64 72 62 2f 64 72 62 2e y/1.9.1/drbrb:01f0 72 62 3a 31 35 34 35 3a 69 6e 20 60 73 65 74 75 rb:1545:in setu
0200 70 5f 6d 65 73 73 61 67 65 27 22 45 2f 6f 70 74 p_message""E/opt
0210 2f 72 75 62 79 2d 31 2e 39 2e 31 2d 70 32 34 33 /ruby-1.9.1-p243
0220 2f 6c 69 62 2f 72 75 62 79 2f 31 2e 39 2e 31 2f /lib/ruby/1.9.1/
0230 64 72 62 2f 64 72 62 2e 72 62 3a 31 34 39 37 3a drbrb:1497:
0240 69 6e 20 60 70 65 72 66 6f 72 6d 27 22 5b 2f 6f in perform""/o 0250 70 74 2f 72 75 62 79 2d 31 2e 39 2e 31 2d 70 32
pt/ruby-1.9.1-p2 0260 34 33 2f 6c 69 62 2f 72 75 62 79 2f 31 2e 39 2e 43/lib/ruby/1.9. 0270 31 2f 64 72 62 2f 64 72 62 2e 72 62 3a
31 35 39 1/drbrb:159 0280 32 3a 69 6e 20 60 62 6c 6f 63 6b 20 28 32 20 6c 2:in block (2 l
0290 65 76 65 6c 73 29 20 69 6e 20 6d 61 69 6e 5f 6c evels) in main_l
02a0 6f 6f 70 27 22 42 2f 6f 70 74 2f 72 75 62 79 2d oop""B/opt/ruby-
02b0 31 2e 39 2e 31 2d 70 32 34 33 2f 6c 69 62 2f 72 1.9.1-p243/lib/r
02c0 75 62 79 2f 31 2e 39 2e 31 2f 64 72 62 2f 64 72 uby/1.9.1/drbrdr
02d0 62 2e 72 62 3a 31 35 38 38 3a 69 6e 20 60 6c 6f b.rb:1588:in lo 02e0 6f 70 27 22 50 2f 6f 70 74 2f 72 75 62 79 2d 31
op""P/opt/ruby-1 02f0 2e 39 2e 31 2d 70 32 34 33 2f 6c 69 62 2f 72 75 .9.1-p243/lib/ru 0300 62 79 2f 31 2e 39 2e 31 2f 64 72 62 2f
64 72 62 by/1.9.1/drbrb:0310 2e 72 62 3a 31 35 38 38 3a 69 6e 20 60 62 6c 6f .rb:1588:in blo
0320 63 6b 20 69 6e 20 6d 61 69 6e 5f 6c 6f 6f 70 27 ck in main_loop'

With the attached patch, which makes the server not send any data when the client tries to close the connection:

No.	Time	Source	Destination	Protocol	Info
1	0.000000	::1	::1	TCP	55904 > msgsrvr [SYN] Seq=0 Win=32752 Len=0 MSS=16376 TSV=112384805 TSER=0 WS=7
2	0.000014	::1	::1	TCP	msgsrvr > 55904 [SYN, ACK] Seq=0 Ack=1 Win=32728 Len=0 MSS=16376 TSV=112384805 TSER=112384805 WS=7
3	0.000025	::1	::1	TCP	55904 > msgsrvr [ACK] Seq=1 Ack=1 Win=32768 Len=0 TSV=112384805 TSER=112384805
4	0.000212	::1	::1	TCP	55904 > msgsrvr [PSH, ACK] Seq=1 Ack=1 Win=32768 Len=68 TSV=112384805 TSER=112384805
5	0.000220	::1	::1	TCP	msgsrvr > 55904 [ACK] Seq=1 Ack=69 Win=32768 Len=0 TSV=112384805 TSER=112384805
6	0.000655	::1	::1	TCP	msgsrvr > 55904 [PSH, ACK] Seq=1 Ack=69 Win=32768 Len=45 TSV=112384805 TSER=112384805
7	0.000663	::1	::1	TCP	55904 > msgsrvr [ACK] Seq=69 Ack=46 Win=32768 Len=0 TSV=112384805 TSER=112384805
8	0.001523	::1	::1	TCP	55904 > msgsrvr [FIN, ACK] Seq=69 Ack=46 Win=32768 Len=0 TSV=112384805 TSER=112384805
9	0.001685	::1	::1	TCP	msgsrvr > 55904 [FIN, ACK] Seq=46 Ack=70 Win=32768 Len=0 TSV=112384805 TSER=112384805
10	0.001700	::1	::1	TCP	55904 > msgsrvr [ACK] Seq=70 Ack=47 Win=32768 Len=0 TSV=112384805 TSER=112384805
=end					

Associated revisions

Revision d0ed935d5bf8c3fce9800742a36e44fb7f63dda4 - 10/14/2019 05:20 AM - jeremyevans (Jeremy Evans)

Fix some DRb issues (#2552)

- Handle BasicObject in drb

Also fix a bug in rescue clause of any_to_s because sprintf does not handle the %l modifier.

Fixes [Bug #7833]

- Do not send a reply to the client if there is a connection error

This allows for normal TCP shutdown (fin-ack-fin-ack instead of fin-ack-push-rst).

Patch from pierre@mouraf.org (Pierre-Alexandre Meyer).

Fixes [Bug #2339]

- Detect fork and do not reuse forked connections in drb

This associates each DRbConn with a pid, and if the pid changes, it closes any DRbConns in the pool with a pid that no longer matches. This fixes DRb servers from sending messages intended for one client to another client after forking.

Fixes [Bug #2718]

Fixes [Bug #14471]

Revision d0ed935d5bf8c3fce9800742a36e44fb7f63dda4 - 10/14/2019 05:20 AM - jeremyevans (Jeremy Evans)

Fix some DRb issues (#2552)

- Handle BasicObject in drb

Also fix a bug in rescue clause of any_to_s because sprintf does not handle the %l modifier.

Fixes [Bug #7833]

- Do not send a reply to the client if there is a connection error

This allows for normal TCP shutdown (fin-ack-fin-ack instead of fin-ack-push-rst).

Patch from pierre@mouraf.org (Pierre-Alexandre Meyer).

Fixes [Bug #2339]

- Detect fork and do not reuse forked connections in drb

This associates each DRbConn with a pid, and if the pid changes, it closes any DRbConns in the pool with a pid that no longer matches. This fixes DRb servers from sending messages intended for one client to another client after forking.

Fixes [Bug #2718]

Fixes [Bug #14471]

Revision d0ed935d - 10/14/2019 05:20 AM - jeremyevans (Jeremy Evans)

Fix some DRb issues (#2552)

- Handle BasicObject in drb

Also fix a bug in rescue clause of any_to_s because sprintf does not handle the %l modifier.

Fixes [Bug #7833]

- Do not send a reply to the client if there is a connection error

This allows for normal TCP shutdown (fin-ack-fin-ack instead of fin-ack-push-rst).

Patch from pierre@mouraf.org (Pierre-Alexandre Meyer).

Fixes [Bug #2339]

- Detect fork and do not reuse forked connections in drb

This associates each DRbConn with a pid, and if the pid changes, it closes any DRbConns in the pool with a pid that no longer matches. This fixes DRb servers from sending messages intended for one client to another client after forking.

History

#1 - 11/26/2009 12:08 PM - **ujihisa (Tatsuhiko Ujihisa)**

- Status changed from Open to Assigned
- Assignee set to *seki (Masatoshi Seki)*

=begin
=end

#2 - 08/27/2019 04:53 PM - **jeremyevans0 (Jeremy Evans)**

- Project changed from 10 to Ruby
- Description updated
- Backport set to 2.5: UNKNOWN, 2.6: UNKNOWN

#3 - 08/27/2019 06:20 PM - **jeremyevans0 (Jeremy Evans)**

- File *drb-no-reply-on-connerror-2339.patch* added

This issue still occurs in the master branch. The originally submitted patch needs a small change to apply. An updated patch is attached.

#4 - 10/14/2019 05:20 AM - **jeremyevans (Jeremy Evans)**

- Status changed from Assigned to Closed

Applied in changeset [gitId0ed935d5bf8c3fce9800742a36e44fb7f63dda4](#).

Fix some DRb issues ([#2552](#))

- Handle BasicObject in drb

Also fix a bug in rescue clause of any_to_s because sprintf does not handle the %l modifier.

Fixes [Bug [#7833](#)]

- Do not send a reply to the client if there is a connection error

This allows for normal TCP shutdown (fin-ack-fin-ack instead of fin-ack-push-rst).

Patch from pierre@mouraf.org (Pierre-Alexandre Meyer).

Fixes [Bug [#2339](#)]

- Detect fork and do not reuse forked connections in drb

This associates each DRbConn with a pid, and if the pid changes, it closes any DRbConns in the pool with a pid that no longer matches. This fixes DRb servers from sending messages intended for one client to another client after forking.

Fixes [Bug [#2718](#)]
Fixes [Bug [#14471](#)]

Files

client.rb	371 Bytes	11/06/2009	pierre@mouraf.org (Pierre-Alexandre Meyer)
server.rb	393 Bytes	11/06/2009	pierre@mouraf.org (Pierre-Alexandre Meyer)
drb.rb.patch	469 Bytes	11/06/2009	pierre@mouraf.org (Pierre-Alexandre Meyer)
drb-no-reply-on-connerror-2339.patch	1.07 KB	08/27/2019	jeremyevans0 (Jeremy Evans)