Ruby master - Bug #13864

Rinda multicast test failures due to missing default route

09/04/2017 01:47 PM - vo.x (Vit Ondruch)

Status:	Assigned		
Priority:	Normal		
Assignee:	seki (Masatoshi Seki)		
Target version:			
ruby -v:	ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-linux]	**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

Trying to build Ruby for Fedora, we are using tool called mock 1. This tools allows to execute the build in container, using systemd-nspawn on background 2 and this in turn allows the container to be isolated from network 3. However, in this setup, there are some Rinda multicast test failures:

```
$ sudo systemd-nspawn '-D' '/var/lib/mock/fedora-rawhide-x86_64/root' --private-network
Spawning container root on /var/lib/mock/fedora-rawhide-x86_64/root.
Press ^] three times within 1s to kill container.
-bash: cannot set terminal process group (-1): Inappropriate ioctl for device
-bash: no job control in this shell
-bash-4.4# su - mockbuild
-bash: cannot set terminal process group (1): Inappropriate ioctl for device
-bash: no job control in this shell
[mockbuild@root ~]$ cd /mnt/
[mockbuild@root mnt]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
[mockbuild@root ~]$ ip route
[mockbuild@root mnt]$
Run options:

# Running tests:

[ 1/39] Rinda::TestRingFinger#test_make_socket_ipv4_multicast = 0.00 s
  1) Error:
Rinda::TestRingFinger#test_make_socket_ipv4_multicast:
Errno::ENETUNREACH: Network is unreachable - connect(2) for 239.0.0.1:7647
    /usr/share/ruby/rinda/ring.rb:431:in `connect'
    /usr/share/ruby/rinda/ring.rb:431:in `make_socket'
    test/rinda/test_rinda.rb:804:in `test_make_socket_ipv4_multicast'

[ 2/39] Rinda::TestRingFinger#test_make_socket_ipv4_multicast_hops = 0.00 s
  2) Error:
Rinda::TestRingFinger#test_make_socket_ipv4_multicast_hops:
Errno::ENETUNREACH: Network is unreachable - connect(2) for 239.0.0.1:7647
    /usr/share/ruby/rinda/ring.rb:431:in `connect'
    /usr/share/ruby/rinda/ring.rb:431:in `make_socket'
    test/rinda/test_rinda.rb:821:in `test_make_socket_ipv4_multicast_hops'

[ 8/39] Rinda::TestRingServer#test_make_socket_ipv4_multicast = 0.00 s
  3) Error:
Rinda::TestRingServer#test_make_socket_ipv4_multicast:
Errno::ENODEV: No such device - setsockopt(2)
    /usr/share/ruby/rinda/ring.rb:154:in `setsockopt'
    /usr/share/ruby/rinda/ring.rb:154:in `make_socket'
    test/rinda/test_rinda.rb:647:in `test_make_socket_ipv4_multicast'

[11/39] Rinda::TestRingServer#test_ring_server_ipv4_multicast = 0.00 s
```

```
  4) Error:
Rinda::TestRingServer#test_ring_server_ipv4_multicast:
Errno::ENODEV: No such device - setsockopt(2)
    /usr/share/ruby/rinda/ring.rb:154:in `setsockopt'
    /usr/share/ruby/rinda/ring.rb:154:in `make_socket'
    /usr/share/ruby/rinda/ring.rb:108:in `block in initialize'
    /usr/share/ruby/rinda/ring.rb:106:in `each'
    /usr/share/ruby/rinda/ring.rb:106:in `initialize'
    test/rinda/test_rinda.rb:689:in `new'
    test/rinda/test_rinda.rb:689:in `test_ring_server_ipv4_multicast'

Leaked file descriptor: Rinda::TestRingServer#test_ring_server_ipv4_multicast: 10 : #<Socket:fd 10
>
Closed file descriptor: Rinda::TestRingServer#test_ring_server_ipv6_multicast: 10
Finished tests in 0.977804s, 39.8853 tests/s, 266.9247 assertions/s.
39 tests, 261 assertions, 0 failures, 4 errors, 4 skips

ruby -v: ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-linux]
```

As you can see from the output, there is only loopback device and no default route specified. If I added the default route via ip route add default via 127.0.0.1, the tests would succeed.

Now, there are several possibilities to narrow this, not really sure which I should choose:

1. Just disable the tests in Fedora.
2. Add the default route.
3. It could be possible to detect this scenario and skip the tests either by the test suite or by the build script.
4. The Rinda::RingServer#make_socket actually allows to specify interface, so if that is explicitly specified to '127.0.0.1', the test could be made to pass, but Rinda::RingFinger#make_socket does not support this :/

I should also note, that it seems that systemd guys don't intend to add the default route while mock upstream considers the option to make it available 4, but this involves some hackery. So I thought I'll ask here ...

History

#1 - 09/04/2017 08:49 PM - shevegen (Robert A. Heiler)

Well no surprise - the systemd guys know better than any standard ever written, too. ;)

I think 1) does not make a lot of sense if it is code within ruby that fails.

I actually thought that Rinda is abandoned since a long time... last time I heard about it
was in 1.8.x but admittedly I did not keep up with everything ...

Recent stdlib does have it though so it is part of the official ruby distribution:

https://ruby-doc.org/stdlib-2.4.1/libdoc/rinda/rdoc/Rinda.html

I think option 3) also sounds a bit like option 1).

2) and 4) seem better.

For http://www.rubydoc.info/stdlib/rinda/Rinda%2FRingFinger:make_socket how would the
API have to be changed to make the tests pass? It should be easy to test for localhost
and change any API to support it but admittedly I dont know much at all about Rinda.

#2 - 09/05/2017 12:21 AM - hsbt (Hiroshi SHIBATA)

- Assignee set to seki (Masatoshi Seki)

- Status changed from Open to Assigned

#3 - 09/07/2017 07:10 AM - vo.x (Vit Ondruch)

shevegen (Robert A. Heiler) wrote:

> Well no surprise - the systemd guys know better than any standard ever written, too. ;)

Honestly, this might happen for Docker, real/virtual machines without network interface, real/virtual machines without DHCP etc. It is just coincidence that we stumbled upon this due to systemd-nspawn ...

> For http://www.rubydoc.info/stdlib/rinda/Rinda%2FRingFinger:make_socket how would the API have to be changed to make the tests pass?

It should be similar to http://www.rubydoc.info/stdlib/rinda/Rinda/RingServer#make_socket-instance_method, e.g. #make_socket(address, interface_address = nil)