

Ceph - Feature #15308

LIBRADOS modify Pipe::connect() to return the error code

03/29/2016 03:04 PM - Vikhyat Umrao

Status:	Resolved	Start date:	06/15/2016
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Source:	Support	Reviewed:	
Tags:		Affected Versions:	
Backport:	jewel	Pull request ID:	

Description

LIBRADOS modify Pipe::connect() to return the error code

Steps to Reproduce:

1. create a qemu instance
2. reduce the fd limits with prlimit command
3. try to attach rbd image to this qemu instance
4. ceph client side log file will have logs as given below :

```
7f0b769e7700 4- 192.168.128.30:0/2021513 >> 192.168.128.35:6800/24374 pipe(0x7f0bcabc0000 sd=-1 :0 s=1 pgs=0 cs=0 l=1 c=0x7f0bc55e1ce0).connect couldn't created socket (24) Too many open files
```

5. But qemu logs will not log anything regarding this error as rados_connect call from qemu wont have proper return code from librados Pipe::connect().

Ceph source file : src/msg/simple/Pipe.cc

```
870 int Pipe::connect()
871 {
872     bool got_bad_auth = false;
873
874     ldout(msgr->cct,10) << "connect " << connect_seq << endl;
875     assert(pipe_lock.is_locked());
876
877     .....
878     .....
879
901     // create socket?
902     sd = ::socket(peer_addr.get_family(), SOCK_STREAM, 0);
903     if (sd < 0) {
904         lderr(msgr->cct) << "connect couldn't created socket " << cpp_strerror(errno) << endl;
905         goto fail;
906     }
907
908     .....
909     .....
910
1215 fail:
1216     if (conf->ms_inject_internal_delays) {
1217         ldout(msgr->cct, 10) << " sleep for " << msgr->cct->_conf->ms_inject_internal_delays << d
1218         endl;
1219         utime_t t;
1220         t.set_from_double(msgr->cct->_conf->ms_inject_internal_delays);
1221         t.sleep();
1222     }
1223     pipe_lock.Lock();
```

```

1224 fail_locked:
1225     if (state == STATE_CONNECTING)
1226         fault();
1227     else
1228         ldout(msgr->cct, 3) << "connect fault, but state = " << get_state_name()
1229             << " != connecting, stopping" << endl;
1230
1231 stop_locked:
1232     delete authorizer;
1233     return -1;
1234 }

```

- "connect couldn't created socket (24) Too many open files"
- This error is coming from this function socket() from Pipe::connect() function.
- Here we are going to **fail** label but if we check **fail** we returning "-1" but we need proper error handling in this function.

Subtasks:

Feature # 16310: Take pipe::connect() returned errno to rados_connect() Part2 of http://... **New**

Related issues:

Duplicated by Ceph - Bug #17573: librados doesn't properly report failure to ...	Duplicate	10/13/2016
Copied to Ceph - Backport #17377: jewel: LIBRADOS modify Pipe::connect() to r...	Resolved	

History

#1 - 03/29/2016 03:05 PM - Vikhyat Umrao

More information : https://bugzilla.redhat.com/show_bug.cgi?id=1321570

#2 - 03/29/2016 03:06 PM - Vikhyat Umrao

- Description updated

#3 - 03/29/2016 03:11 PM - Vikhyat Umrao

- Source changed from other to Support

#4 - 03/30/2016 12:00 PM - Vikhyat Umrao

- Assignee deleted (Vikhyat Umrao)

#5 - 03/30/2016 01:41 PM - Vikhyat Umrao

Does async messenger have this problem?

async has this problem as well, see AsyncConnection::_process_connection(), and search for "sd = net.nonblock_connect(get_peer_addr());".

Probably specifying a function-local error code and making sure that gets set everywhere before jumping through to the failure handling.

yeah, we need to save the errno somewhere and assign it to rval of the context of the request.

#6 - 03/30/2016 01:46 PM - Vikhyat Umrao

Thanks Kefu for your inputs and nice discussion.

- Now we need to fix both the messengers : simple messenger and async messenger connect function definitions.

- Then I and Kefu discussed in detail more about it and we concluded we need to fix all the caller functions of these functions with proper return codes.

- I checked parent AsyncConnection::_process_connection() parent functions and they look good.

Like :_process_connection() calls -> nonblock_connect() calls -> generic_connect() -> and generic connect definition is proper.

```
int NetHandler::generic_connect(const entity_addr_t& addr, bool nonblock)
{
    int ret;
    int s = create_socket(addr.get_family());
    if (s < 0)
        return s;

    if (nonblock) {
        ret = set_nonblock(s);
        if (ret < 0) {
            close(s);
            return ret;
        }
    }

    set_socket_options(s);

    ret = ::connect(s, (sockaddr*)&addr.addr, addr.addr_size());
    if (ret < 0) {
        if (errno == EINPROGRESS && nonblock)
            return s;

        ldout(cct, 10) << __func__ << " connect: " << strerror(errno) << endl;
        close(s);
        return -errno;
    }

    return s;
}
```

#7 - 04/01/2016 03:01 PM - Vikhyat Umrao

Hello Josh,

Thanks for your help and nice discussion.

As discussed I have sent first part of pipe::connect() fix : <https://github.com/ceph/ceph/pull/8416>

Will work on qemu patch for fixing error number function fix.

#8 - 04/22/2016 01:13 PM - Vikhyat Umrao

qemu-kvm-rhev RFE Bugzilla : https://bugzilla.redhat.com/show_bug.cgi?id=1329641

#9 - 04/30/2016 01:50 AM - Sage Weil

- Status changed from New to Resolved

#10 - 09/21/2016 08:43 PM - Samuel Just

- Status changed from Resolved to Pending Backport

- Backport set to jewel

#11 - 09/22/2016 12:14 PM - Nathan Cutler

- Copied to Backport #17377: jewel: LIBRADOS modify Pipe::connect() to return the error code added

#12 - 10/14/2016 05:04 PM - Nathan Cutler

- Status changed from Pending Backport to Resolved

#13 - 02/13/2017 05:56 PM - Josh Durgin

- Duplicated by Bug #17573: librados doesn't properly report failure to create socket added