Ceph - Bug #14499

librados rados_monitor_log() aborts if called prior to rados_connect()

01/25/2016 03:23 PM - David Disseldorp

Status:	Resolved	**Start date:**	01/25/2016
Priority:	Low	**Due date:**	
Assignee:	David Disseldorp	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Source:	Community (dev)	**Reviewed:**	
Tags:		**Affected Versions:**	v9.2.1
Backport:		**ceph-qa-suite:**	rados
Regression:	No	**Pull request ID:**	
Severity:	3 - minor		

Description

rados_moditor_log() must be called with a connected rados cluster context, otherwise the calling process may abort with:

#0 0x00007fffe3df8d38 in raise () from /lib64/libc.so.6
#1 0x00007fffe3dfa18a in abort () from /lib64/libc.so.6
#2 0x00007fffe4fe4f4b in ceph::__ceph_assert_fail (assertion=assertion@entry=0x7fffe52b0a9e "monmap.size() > 0", file=file@entry=0x7fffe52b0a44 "mon/MonClient.cc", line=line@entry=580, func=func@entry=0x7fffe52b14c0 <MonClient::_pick_random_mon[abi:cxx11]()::__PRETTY_FUNCTION__> "std::__cxx11::string MonClient::_pick_random_mon()") at common/assert.cc:78
#3 0x00007fffe506930e in MonClient::_pick_random_mon[abi:cxx11]() (this=this@entry=0x77f2c0) at mon/MonClient.cc:580
#4 0x00007fffe507005f in MonClient::_reopen_session (this=this@entry=0x77f2c0, rank=rank@entry=-1, name="") at mon/MonClient.cc:605
#5 0x00007fffe5076d3d in MonClient::_reopen_session (this=this@entry=0x77f2c0) at mon/MonClient.h:190
#6 0x00007fffe506f684 in MonClient::_renew_subs (this=this@entry=0x77f2c0) at mon/MonClient.cc:769
#7 0x00007fffe4f212f1 in renew_subs (this=0x77f880) at ./mon/MonClient.h:260
#8 librados::RadosClient::monitor_log (this=this@entry=0x77f2a0, level="debug", cb=cb@entry=0x46aa70 <_fio_rbd_log_cb>, arg=arg@entry=0x0) at librados/RadosClient.cc:866
#9 0x00007fffe4f0140f in rados_monitor_log (cluster=0x77f2a0, level=level@entry=0x47b89a "debug", cb=cb@entry=0x46aa70 <_fio_rbd_log_cb>, arg=arg@entry=0x0) at librados/librados.cc:2866
#10 0x000000000046b248 in _fio_rbd_connect (td=0x7fffd2e06000) at engines/rbd.c:137
#11 fio_rbd_setup (td=0x7fffd2e06000) at engines/rbd.c:470
#12 0x0000000000433752 in setup_files (td=td@entry=0x7fffd2e06000) at filesetup.c:799
#13 0x000000000045e360 in run_threads (sk_out=sk_out@entry=0x0) at backend.c:2067
#14 0x000000000045e6ed in fio_backend (sk_out=sk_out@entry=0x0) at backend.c:2381
#15 0x000000000040e818 in main (argc=2, argv=0x7fffffffe308, envp=<optimized out>) at fio.c:63

```
578 string MonClient::_pick_random_mon()
579 {
580   assert(monmap.size() > 0);
^^^^-------- here
581   if (monmap.size() == 1) {
582     return monmap.get_name(0);
583   } else {
584     int max = monmap.size();
```

This backtrace was obtained from fio, with the following modification:

```
diff --git a/engines/rbd.c b/engines/rbd.c
index 8252d27..9cc050f 100644
--- a/engines/rbd.c
+++ b/engines/rbd.c
@ -106,6 +106,16 @ failed:

}
```

```
void _fio_rbd_log_cb(void *arg,
const char *line,
+                   const char *who,
+                   uint64_t sec, uint64_t nsec,
+                   uint64_t seq, const char *level,
+                   const char *msg)
{
log_err(msg);
}

static int _fio_rbd_connect(struct thread_data *td)  {
struct rbd_data *rbd = td->io_ops->data;
@ -124,6 +134,12 @ static int _fio_rbd_connect(struct thread_data *td)
goto failed_early;
}

+       r = rados_monitor_log(rbd->cluster, "error", _fio_rbd_log_cb, NULL);
+       if (r < 0) {
+               log_err("rados_monitor_log failed.\n");
+               goto failed_shutdown;
+       }
+
r = rados_connect(rbd->cluster);
if (r < 0) {
log_err("rados_connect failed.\n");

==================================
```

Aborting from a library rather than returning an error (-ENOTCONN) up to the caller is generally considered bad form.
This issue should be easily addressable by checking that (state == CONNECTED) in the librados::RadosClient::monitor_log() code path.

Associated revisions

Revision 5dfcdf69 - 02/09/2016 06:23 PM - David Disseldorp

librados: check connection state in rados_monitor_log

rados_monitor_log() may abort the calling process if called with a
disconnected rados cluster context. Return -ENOTCONN up to the caller,
rather than aborting in such cases.

Fixes: #14499
Signed-off-by: David Disseldorp <ddiss@suse.de>

History

#1 - 01/25/2016 04:28 PM - David Disseldorp

Fix submitted via:
https://github.com/ceph/ceph/pull/7350

#2 - 01/25/2016 07:19 PM - Nathan Cutler

- Status changed from New to Need Review

- Source changed from other to Community (dev)

#3 - 11/14/2016 03:24 PM - David Disseldorp

This bug can be closed - the fix was merged long ago.

#4 - 11/15/2016 09:30 PM - Nathan Cutler

- Status changed from Need Review to Resolved