rbd - Bug #11537

librbd: crash when two clients try to write to an exclusive locked image

05/05/2015 11:15 PM - Josh Durgin

Status:	Resolved	**% Done:**	0%
Priority:	Urgent	**Spent time:**	0.00 hour
Assignee:	Jason Dillaman		
Category:			
Target version:			
Source:	Development	**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:	hammer	**Pull request ID:**	
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	
Reviewed:			

Description

Using two instances of rbd bench-write on the same image at the same time:

```
term1 $ rbd create --image-feature exclusive-lock -s 100 foo
term1 $ rbd bench-write foo
term2 $ rbd bench-write foo
```

Results in the first instance crashing:

```
#0  0x00007fadbc6dae2b in raise (sig=<value optimized out>) at ../nptl/sysdeps/unix/sysv/linux/pt-raise.c:41
#1  0x00000000008fac1b in reraise_fatal (signum=6) at global/signal_handler.cc:59
#2  0x00000000008faf74 in handle_fatal_signal (signum=6) at global/signal_handler.cc:109
#3  <signal handler called>
#4  0x00007fadbb125165 in *__GI_raise (sig=<value optimized out>) at ../nptl/sysdeps/unix/sysv/linux/raise.c:64
#5  0x00007fadbb127f70 in *__GI_abort () at abort.c:92
#6  0x00007fadbb9b8dc5 in __gnu_cxx::__verbose_terminate_handler() () from /usr/lib/libstdc++.so.6
#7  0x00007fadbb9b7166 in ?? () from /usr/lib/libstdc++.so.6
#8  0x00007fadbb9b7193 in std::terminate() () from /usr/lib/libstdc++.so.6
#9  0x00007fadbb9b728e in __cxa_throw () from /usr/lib/libstdc++.so.6
#10 0x00007fadbf71a99d in ceph::__ceph_assert_fail (assertion=0x7fadbfab06e3 "r == 0", file=0x7fadbfab06bf "./common/RWLock.h",
    line=71, func=0x7fadbfab0c10 "void RWLock::get_read() const") at common/assert.cc:77
#11 0x00007fadbf517e77 in RWLock::get_read (this=0x7fadb0001568) at common/RWLock.h:71
#12 0x00007fadbf517fa6 in RWLock::RLocker::RLocker(RWLock const&) () from /home/joshd/ceph/src/.libs/librbd.so.1
#13 0x00007fadbf5168d5 in librbd::AbstractWrite::send_pre (this=0x7fadb24c9a10) at librbd/AioRequest.cc:437
#14 0x00007fadbf516864 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#15 0x00007fadbf59a126 in librbd::LibrbdWriteback::write (this=0x7fadb0009440, oid=..., oloc=..., off=1900544, len=1040384,
    snapc=..., bl=..., mtime=..., trunc_size=0, trunc_seq=0, oncommit=0x7fadb16b7a10) at librbd/LibrbdWriteback.cc:168
#16 0x00007fadbfa4e9af in ObjectCacher::bh_write (this=0x7fadb000b140, bh=0x4b3a8f0) at osdc/ObjectCacher.cc:847
#17 0x00007fadbfa56bd0 in ObjectCacher::flush_set (this=0x7fadb000b140, oset=0x7fadb0009ad0, onfinish=0x7fadb0249600)
    at osdc/ObjectCacher.cc:1722
```

```
#18 0x00007fadbf5378d8 in librbd::ImageCtx::flush_cache_aio (this=0x7fadb00013d0, onfinish=0x7fadb
0249600)
    at librbd/ImageCtx.cc:617
#19 0x00007fadbf537991 in librbd::ImageCtx::flush_cache (this=0x7fadb00013d0) at librbd/ImageCtx.c
c:627
#20 0x00007fadbf586862 in librbd::_flush (ictx=0x7fadb00013d0) at librbd/internal.cc:3719
#21 0x00007fadbf54adbf in librbd::ImageWatcher::release_lock (this=0x7fadb000b6f0) at librbd/Image
Watcher.cc:382
#22 0x00007fadbf54cf12 in librbd::ImageWatcher::notify_release_lock (this=0x7fadb000b6f0) at librb
d/ImageWatcher.cc:583
#23 0x00007fadbf561dcf in boost::_mfi::mf0<void, librbd::ImageWatcher>::operator()(librbd::ImageWa
tcher*) const ()
   from /home/joshd/ceph/src/.libs/librbd.so.1
#24 0x00007fadbf55fbee in void boost::_bi::list1<boost::_bi::value<librbd::ImageWatcher*> >::opera
tor()<boost::_mfi::mf0<void, librbd::ImageWatcher>, boost::_bi::list1<int&> >(boost::_bi::type<voi
d>, boost::_mfi::mf0<void, librbd::ImageWatcher>&, boost::_bi::list1<int&>&, int) () from /home/jo
shd/ceph/src/.libs/librbd.so.1
#25 0x00007fadbf55d6e0 in void boost::_bi::bind_t<void, boost::_mfi::mf0<void, librbd::ImageWatche
r>, boost::_bi::list1<boost::_bi::value<librbd::ImageWatcher*> > >::operator()<int>(int&) () from
/home/joshd/ceph/src/.libs/librbd.so.1
#26 0x00007fadbf55ae45 in boost::detail::function::void_function_obj_invoker1<boost::_bi::bind_t<v
oid, boost::_mfi::mf0<void, librbd::ImageWatcher>, boost::_bi::list1<boost::_bi::value<librbd::Ima
geWatcher*> > >, void, int>::invoke(boost::detail::function::function_buffer&, int) () from /home/
joshd/ceph/src/.libs/librbd.so.1
#27 0x00007fadbf518df1 in boost::function1<void, int>::operator()(int) const () from /home/joshd/c
eph/src/.libs/librbd.so.1
#28 0x00007fadbf5183f6 in FunctionContext::finish(int) () from /home/joshd/ceph/src/.libs/librbd.s
o.1
#29 0x00007fadbf5110e5 in Context::complete(int) () from /home/joshd/ceph/src/.libs/librbd.so.1
#30 0x00007fadbf564709 in librbd::TaskFinisher<librbd::ImageWatcher::Task>::complete(librbd::Image
Watcher::Task const&) ()
   from /home/joshd/ceph/src/.libs/librbd.so.1
#31 0x00007fadbf564614 in librbd::TaskFinisher<librbd::ImageWatcher::Task>::C_Task::finish(int) ()
   from /home/joshd/ceph/src/.libs/librbd.so.1
#32 0x00007fadbf5110e5 in Context::complete(int) () from /home/joshd/ceph/src/.libs/librbd.so.1
#33 0x00007fadbf71bc77 in Finisher::finisher_thread_entry (this=0x7fadb000b9d0) at common/Finisher
.cc:59
#34 0x00007fadbf53b8b4 in Finisher::FinisherThread::entry() () from /home/joshd/ceph/src/.libs/lib
rbd.so.1
#35 0x00007fadbf61bcca in Thread::entry_wrapper (this=0x7fadb000bae8) at common/Thread.cc:84
#36 0x00007fadbf61bc18 in Thread::_entry_func (arg=0x7fadb000bae8) at common/Thread.cc:66
#37 0x00007fadbc6d28ba in start_thread (arg=<value optimized out>) at pthread_create.c:300
#38 0x00007fadbb1c202d in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:112
#39 0x0000000000000000 in ?? ()
```

This may be fixed by the locking cleanups already. But the 2nd instance hangs, not making progress instead of breaking the lock.

Related issues:

Copied to rbd - Backport #12235: librbd: crash when two clients try to write ...	**Resolved**	**05/05/2015**

History

#1 - 05/06/2015 07:06 PM - Jason Dillaman

- Status changed from New to In Progress

- Assignee set to Jason Dillaman

#2 - 05/15/2015 03:23 PM - Jason Dillaman

- Status changed from In Progress to Fix Under Review

master PR: https://github.com/ceph/ceph/pull/4695

#3 - 06/22/2015 05:22 PM - Jason Dillaman

- Status changed from Fix Under Review to Pending Backport

- Backport set to hammer

Similar issue occurs in Hammer -- causes deadlock instead of a crash due to differences in locking addressed in PR4528

#4 - 08/30/2015 01:57 PM - Loïc Dachary

- Status changed from Pending Backport to Resolved