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

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  0x00000000000fac1b in reraise_fatal (signum=6) at global/signal_handler.cc:59
#2  0x00000000000fa7f4 in handle_fatal_signal (signum=6) at global/signal_handler.cc:109
#3  <signal handler called>
#4  0x0000000fadb125165 in __GI_raise (sig=<value optimized out>) at ../nptl/sysdeps/unix/sysv/linux/raise.c:64
#5  0x0000000fadb12f770 in __GI_abrt () at abort.c:92
#6  0x0000000fadb17b619 in __gnu_cxx::__verbose_terminate_handler() from /usr/lib/libstdc++.so.6
#7  0x0000000fadb17b616 in __cxa_throw () from /usr/lib/libstdc++.so.6
#8  0x0000000fadb17b619 in std::terminate() from /usr/lib/libstdc++.so.6
#9  0x0000000fadb17b61e in __cxa_rethrow () from /usr/lib/libstdc++.so.6
#10 0x0000000fadb19a9d in ceph::__ceph_assert_fail (assertion=0x7fadbfbab06e3 "r == 0", file=0x7fadbfbab06bf "./common/RWLock.h", line=71, func=0x7fadbfbab06e3 void RWLock::get_read() const") at common/assert.cc:77
#11 0x0000000fadb1f77e77 in RWLock::get_read (this=0x7fadb0001568) at common/RWLock.h:71
#12 0x0000000fadb1f7fa6 in RWLock::RLocker::RLocker(RWLock const&) () from /home/joshd/ceph/src/objs/librbd.so.1
#13 0x0000000fadb1f6b612 in librbd::AbstractWrite::send_pre (this=0x7fadb24c9a10) at librbd/AioRequest.cc:437
#14 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#15 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#16 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#17 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#18 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#19 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#20 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#21 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#22 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#23 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
#24 0x0000000fadb1f6b612 in librbd::AbstractWrite::send (this=0x7fadb24c9a10) at librbd/AioRequest.cc:426
```
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 Backport #12235: librbd: crash when two clients try to write to an ... 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 Need Review

**master PR:** [https://github.com/ceph/ceph/pull/4695](https://github.com/ceph/ceph/pull/4695)

**#3** - 06/22/2015 05:22 PM - Jason Dillaman
- Status changed from Need 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 - Loic Dachary
- Status changed from Pending Backport to Resolved