

## rbd - Bug #16363

### Journal duplicate op detection can cause lockdep error

06/17/2016 12:50 PM - Jason Dillaman

<b>Status:</b>	Resolved	<b>Start date:</b>	06/17/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jason Dillaman	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour
<b>Source:</b>	other	<b>Reviewed:</b>	
<b>Tags:</b>		<b>Affected Versions:</b>	
<b>Backport:</b>	jewel	<b>ceph-qa-suite:</b>	
<b>Regression:</b>	No	<b>Pull request ID:</b>	
<b>Severity:</b>	3 - minor		

#### Description

```
2016-06-17 11:18:54.385522 7f7bae7fc700 20 JournalMetadata: C_AllocateTag: handle_get_next_tag_tid
: r=0
2016-06-17 11:18:54.385527 7f7bae7fc700 20 JournalMetadata: C_AllocateTag: send_tag_create
2016-06-17 11:18:54.557149 7f7bae7fc700 20 JournalMetadata: C_AllocateTag: handle_tag_create: r=0
2016-06-17 11:18:54.557155 7f7bae7fc700 20 JournalMetadata: C_AllocateTag: send_get_tag
2016-06-17 11:18:54.660836 7f7bae7fc700 20 JournalMetadata: C_AllocateTag: handle_get_tag: r=0
2016-06-17 11:18:54.660849 7f7bae7fc700 20 librbd::Journal: allocated journal tag: tid=4, data=[mi
rror_uuid=c83a3383-60bd-49ad-b76f-7fc196b682b4, predecessor_mirror_uuid=c83a3383-60bd-49ad-b76f-7f
c196b682b4, predecessor_tag_tid=4, predecessor_entry_tid=99]
2016-06-17 11:18:54.660852 7f7bae7fc700 20 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] handle_allocate_local_tag: r=0
2016-06-17 11:18:54.660853 7f7bae7fc700 20 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] process_entry: processing entry tid=108
2016-06-17 11:18:54.660856 7f7bae7fc700 20 librbd::journal::Replay: 0x7f7b400129d0 process: on_rea
dy=0x7f7b5c01c260, on_safe=0x7f7b5c044dd0
2016-06-17 11:18:54.660861 7f7bae7fc700 20 librbd::journal::Replay: 0x7f7b400129d0 handle_event: S
nap remove event
2016-06-17 11:18:54.660863 7f7bae7fc700 -1 librbd::journal::Replay: duplicate op tid detected: 1
2016-06-17 11:18:54.660866 7f7bae7fc700 20 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] handle_process_entry_safe: commit_tid=108, r=-22
2016-06-17 11:18:54.660868 7f7bae7fc700 -1 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] handle_process_entry_safe: failed to commit journal event: (22) Inval
id id argument
2016-06-17 11:18:54.660874 7f7bae7fc700 20 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] handle_replay_complete: r=-22
2016-06-17 11:18:54.660875 7f7bae7fc700 -1 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] handle_replay_complete: replay encountered an error: (22) Invalid arg
ument
2016-06-17 11:18:54.660878 7f7bae7fc700 20 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] set_state_description: -22 failed to commit journal event
2016-06-17 11:18:54.660877 7f7b6d7fa700 20 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] handle_process_entry_ready:
2016-06-17 11:18:54.660881 7f7b6d7fa700 20 rbd::mirror::ImageReplayer: 0x7f7b74007c90 [3/2f6e6235-
b765-470c-a6d2-b9cefeb19988] handle_replay_ready: enter
2016-06-17 11:18:54.660879 7f7bae7fc700 0
-----
existing dependency rbd::mirror::ImageReplayer 4 11362ae8944a (130) -> Replay<I>::m_lock (103) at:

2016-06-17 11:18:54.662296 7f7bae7fc700 0 new dependency Replay<I>::m_lock (103) -> rbd::mirror::
ImageReplayer 4 11362ae8944a (130) creates a cycle at
ceph version 10.2.0-2463-g204a1bd (204a1bd678617ffb33be1a1faed7ee7c01b0f19f)
1: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::set_state_description(int, std::string const&)+
0x78) [0x7f7bd2ca7648]
```

```
2: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::handle_replay_complete(int, std::string const&
+0x34c) [0x7f7bd2ca8d7c]
3: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::handle_process_entry_safe(journal::ReplayEntry
const&, int)+0x247) [0x7f7bd2ca9637]
4: (Context::complete(int)+0x9) [0x7f7bd2c933c9]
5: (librbd::journal::Replay<librbd::ImageCtx>::create_op_context_callback(unsigned long, Context*
, Context*, librbd::journal::Replay<librbd::ImageCtx>::OpEvent**) +0x24b) [0x7f7bd2d8cc5b]
6: (librbd::journal::Replay<librbd::ImageCtx>::handle_event(librbd::journal::SnapRemoveEvent cons
t&, Context*, Context*)+0x98) [0x7f7bd2d8f918]
7: (librbd::journal::Replay<librbd::ImageCtx>::process(ceph::buffer::list::iterator*, Context*, C
ontext*)+0x432) [0x7f7bd2d905e2]
8: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::process_entry()+0xfd) [0x7f7bd2cab3fd]
9: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::handle_allocate_local_tag(int)+0x79) [0x7f7bd2c
ab639]
10: (Context::complete(int)+0x9) [0x7f7bd2c933c9]
11: (() +0x279bb1) [0x7f7bd2d3abb1]
12: (() +0x35e940) [0x7f7bd2e1f940]
13: (() +0xa562d) [0x7f7bc9f8c62d]
14: (() +0x8d0d9) [0x7f7bc9f740d9]
15: (() +0x17d34e) [0x7f7bca06434e]
16: (() +0x80a4) [0x7f7bc96af0a4]
17: (clone()+0x6d) [0x7f7bc801704d]
```

```
2016-06-17 11:18:54.662335 7f7bae7fc700 0 btw, i am holding these locks:
2016-06-17 11:18:54.662336 7f7bae7fc700 0   Replay<I>::m_lock (103)
2016-06-17 11:18:54.662337 7f7bae7fc700 0   librbd::ImageCtx::owner_lock (0x7f7bb4089ca0) (201)
2016-06-17 11:18:54.662338 7f7bae7fc700 0
```

```
2016-06-17 11:18:54.663848 7f7bae7fc700 -1 common/lockdep.cc: In function 'int lockdep_will_lock(c
onst char*, int, bool)' thread 7f7bae7fc700 time 2016-06-17 11:18:54.662339
common/lockdep.cc: 330: FAILED assert(0)
```

```
ceph version 10.2.0-2463-g204a1bd (204a1bd678617ffb33be1a1faed7ee7c01b0f19f)
1: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x82) [0x7f7bd2ed7732]
2: (lockdep_will_lock(char const*, int, bool)+0xde6) [0x7f7bd2f46846]
3: (Mutex::Lock(bool)+0x157) [0x7f7bd2eb80b7]
4: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::set_state_description(int, std::string const&)+
0x78) [0x7f7bd2ca7648]
5: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::handle_replay_complete(int, std::string const&
+0x34c) [0x7f7bd2ca8d7c]
6: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::handle_process_entry_safe(journal::ReplayEntry
const&, int)+0x247) [0x7f7bd2ca9637]
7: (Context::complete(int)+0x9) [0x7f7bd2c933c9]
8: (librbd::journal::Replay<librbd::ImageCtx>::create_op_context_callback(unsigned long, Context*
, Context*, librbd::journal::Replay<librbd::ImageCtx>::OpEvent**) +0x24b) [0x7f7bd2d8cc5b]
9: (librbd::journal::Replay<librbd::ImageCtx>::handle_event(librbd::journal::SnapRemoveEvent cons
t&, Context*, Context*)+0x98) [0x7f7bd2d8f918]
10: (librbd::journal::Replay<librbd::ImageCtx>::process(ceph::buffer::list::iterator*, Context*,
Context*)+0x432) [0x7f7bd2d905e2]
11: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::process_entry()+0xfd) [0x7f7bd2cab3fd]
12: (rbd::mirror::ImageReplayer<librbd::ImageCtx>::handle_allocate_local_tag(int)+0x79) [0x7f7bd2
cab639]
13: (Context::complete(int)+0x9) [0x7f7bd2c933c9]
14: (() +0x279bb1) [0x7f7bd2d3abb1]
15: (() +0x35e940) [0x7f7bd2e1f940]
16: (() +0xa562d) [0x7f7bc9f8c62d]
17: (() +0x8d0d9) [0x7f7bc9f740d9]
18: (() +0x17d34e) [0x7f7bca06434e]
19: (() +0x80a4) [0x7f7bc96af0a4]
20: (clone()+0x6d) [0x7f7bc801704d]
NOTE: a copy of the executable, or `objdump -rDS <executable>` is needed to interpret this.
```

**Related issues:**

Copied to rbd - Backport #16423: jewel: Journal duplicate op detection can ca...

**Resolved**

## History

---

**#1 - 06/17/2016 12:50 PM - Jason Dillaman**

- Backport set to jewel

**#2 - 06/17/2016 01:47 PM - Jason Dillaman**

- Subject changed from *Journal duplicate lock detection can cause lockdep error* to *Journal duplicate op detection can cause lockdep error*

**#3 - 06/20/2016 02:26 PM - Jason Dillaman**

- Status changed from *New* to *In Progress*

- Assignee set to *Jason Dillaman*

**#4 - 06/20/2016 02:34 PM - Jason Dillaman**

- Status changed from *In Progress* to *Need Review*

PR: <https://github.com/ceph/ceph/pull/9819>

**#5 - 06/21/2016 04:52 PM - Mykola Golub**

- Status changed from *Need Review* to *Pending Backport*

**#6 - 06/22/2016 09:08 PM - Nathan Cutler**

- Copied to Backport #16423: *jewel: Journal duplicate op detection can cause lockdep error added*

**#7 - 08/10/2016 07:21 AM - Loic Dachary**

- Status changed from *Pending Backport* to *Resolved*