Linux kernel client - Feature #10585

use new, more reliable version of watch/notify

01/20/2015 09:06 AM - Josh Durgin

Status:	Resolved	**% Done:**	0%
Priority:	Normal	**Spent time:**	0.00 hour
Assignee:	Douglas Fuller		
Category:	libceph		
Target version:	sprint2		
Source:	other	**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:			

Description

The interface exposed by librados has everything that needs to be available to the user and a description of most of the rados-level semantics [1]. Most of this work will be in osd_client, and a little bit to make rbd use it.

In rbd, opening an image non-readonly causes a watch to be established on the header object of the image. For historical reasons, notifications were originally sent with no payload and any notification on the image header resulted in re-reading all the mutable image metadata. In userspace this means incrementing the ImageCtx::refresh_seq counter, which is checked before each operation to see if the image metadata needs to be reread. When a watch is lost, the error callback is called and rbd compensates for possible missed notifications by incrementing refresh_seq to reread the header before the next operation.

In hammer and beyond the notify payload is used by images with the exclusive lock feature bit to proxy management operations to the lock holder, but that's a separate issue. For now the payload can continue being ignored by krbd, and krbd doesn't need to send notifications yet.

These details are handled by ImageWatcher in userspace, in particular see reregister_watch() for watch error handling [2], and how notifications are now explicitly acked (rados_notify_ack()) by rbd.

In terms of the low-level implementation of watch/notify, the usual MOSDOp message for rados operations is used to register/unregister watches and send notifications with watch/notify-specific fields. The client periodically pings osds serving watches to make sure the connection is alive for any osds serving watches [3]. The kernel should already be doing this. What it doesn't do yet is expose when a watch has an error and needs to be reregistered, and the watch flush mechanism may need to change as well. Note that in the userspace analogue of osd_client, the Objecter, watch/notify are called "linger" ops for historical reasons. Objecter::handle_watch_notify() takes care of MWatchNotify [4] messages, which are notifications or watch errors received from the OSD.

[1] https://github.com/ceph/ceph/blob/7e5b81b38106654c0b6760b597058ad6e7655dda/src/include/rados/librados.h#L1869

[2] https://github.com/ceph/ceph/blob/796f810398cc4c828a0047ca7a4cc188a805c2af/src/librbd/ImageWatcher.cc#L987

[3] https://github.com/ceph/ceph/blob/780576ba62a3de8decdedae4545af5a853465738/src/osdc/Objecter.cc#L548

[4] https://github.com/ceph/ceph/blob/889cd874e2ded7a1350659449d777af8f4a7a918/src/messages/MWatchNotify.h

Related issues:

Related to Linux kernel client - Bug #13328: fix notify completion race	**Resolved**	**10/01/2015**
Blocked by Linux kernel client - Feature #9779: libceph: sync up with objecter	**Resolved**	**10/14/2014**

History

#1 - 01/20/2015 09:18 AM - Josh Durgin

- Target version set to sprint2

#2 - 04/28/2015 04:13 PM - Josh Durgin

- Assignee set to Douglas Fuller

#3 - 04/28/2015 04:44 PM - Ilya Dryomov

- Category set to libceph

A high-level discussion with some links:

http://www.spinics.net/lists/ceph-devel/msg21422.html

#4 - 04/28/2015 09:13 PM - Josh Durgin

- Description updated

#5 - 05/07/2015 05:02 PM - Douglas Fuller

- Status changed from New to In Progress

#6 - 06/15/2015 01:44 PM - Douglas Fuller

- Status changed from In Progress to Fix Under Review

#7 - 06/05/2016 07:44 PM - Ilya Dryomov

- Status changed from Fix Under Review to Resolved

Done in 4.7 by way of #9779.