

rbd - Feature #4444

send/receive rbd snapshots

03/14/2013 02:52 PM - Ian Colle

Status: Resolved	% Done: 0%
Priority: Normal	Spent time: 0.00 hour
Assignee: Sage Weil	
Category:	
Target version: v0.61 - Cuttlefish	
Source:	Reviewed:
Tags:	Affected Versions:
Backport:	Pull request ID:
Description	
It would be great to be able to send and receive incremental RBD snapshots as is done in zfs.	
zfs send -i snap1 ... snapN ssh root@host "zfs recieve ..."	
In this case only changes will be sent. This is very useful for creating incremental backups of large volumes.	
Related issues:	
Related to rbd - Feature #4445: librbd: expose changed objects since a given ...	Resolved 10/23/2012
Related to rbd - Feature #4084: rbd: incremental backups	Resolved 10/08/2012
Copied from rbd - Subtask #3272: send/receive rbd snapshots	Rejected 10/08/2012

History

#1 - 03/14/2013 02:53 PM - Ian Colle

- translation missing: en.field_story_points set to 11.00

From Sage - "I think this breaks down into a few different pieces:

- 1) Decide what output format to use. We want to use something that is resembles a portable, standard way of representing an incremental set of changes to a block device (or large file). I'm not sure what is out there, but we should look carefully before making up our own format.
- 2) Expose changes objects between rados snapshots. This is some generic functionality we would bake into librbd that would probably work similarly to how read_iterate() currently does (you specify a callback). We probably also want to provide this information directly to a user, so that they can get a dump of (offsets, length) pairs for integration with their own tool. I expect this is just a core librbd method.
- 3) Write a dumper based on #2 that outputs in format from #1. The callback would (instead of printing file offsets) write the data to the output stream with appropriate metadata indicating which part of the image it is. Ideally the output part would be modular, too, so that we can come back later and implement support for new formats easily. The output data stream should be able to be directed at stdout or a file.
- 4) Write an importer for #1. It would take as input an existing image, assumed to be in the state of the reference snapshot, and write all the changed bits. Take input from stdin or a file.
- 5) If necessary, extend the above so that image resize events are properly handled."

Note this ticket covers 1, 3, and 4 - 2 is 3387.

Estimate 11 points

#2 - 03/14/2013 03:02 PM - Ian Colle

- *Target version set to v0.61 - Cuttlefish*

#3 - 03/14/2013 03:03 PM - Neil Levine

- *Status changed from New to 12*

#4 - 03/23/2013 11:08 AM - Sage Weil

- *Status changed from 12 to In Progress*

#5 - 03/27/2013 11:25 AM - Ian Colle

- *Assignee set to Sage Weil*

#6 - 04/01/2013 11:26 AM - Sage Weil

- *Status changed from In Progress to Resolved*