

rbd - Subtask #3272

Feature # 4084 (Resolved): rbd: incremental backups

send/receive rbd snapshots

10/08/2012 10:32 AM - Pavel Arbuzov

Status: Rejected	% Done: 0%
Priority: Normal	Spent time: 0.00 hour
Assignee:	
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 receive ..."	
In this case only changes will be sent. This is very useful for creating incremental backups of large volumes.	
Related issues:	
Copied to rbd - Feature #4444: send/receive rbd snapshots	Resolved 10/08/2012

History

#1 - 10/23/2012 10:16 AM - Josh Durgin

- Subject changed from send/recieve rbd snapshots to send/receive rbd snapshots

#2 - 01/02/2013 01:16 PM - Sage Weil

- translation missing: en.field_position set to 7

#3 - 02/11/2013 04:13 PM - Josh Durgin

- Parent task set to #4084

#4 - 02/14/2013 05:03 PM - Ian Colle

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

#5 - 03/14/2013 02:57 PM - Ian Colle

- Status changed from New to Rejected

- translation missing: en.field_remaining_hours set to 0.00

#6 - 03/14/2013 02:58 PM - Ian Colle

- Estimated time set to 0.00 h

Moved to 4444 - to track as a feature, not a subtask