Ceph - Bug #10146

ceph-disk: sometimes the journal symlink is not created

11/20/2014 01:35 AM - Dan van der Ster

Status:	Resolved	**% Done:**	0%
Priority:	Normal	**Spent time:**	0.00 hour
Assignee:	Dan van der Ster		
Category:			
Target version:			
Source:	other	**Affected Versions:**	
Tags:		**ceph-qa-suite:**	
Backport:	firefly	**Pull request ID:**	
Regression:	No	**Crash signature (v1):**	
Severity:	3 - minor	**Crash signature (v2):**	
Reviewed:			

Description

Hi,
We observed in practise that sometimes the journal symlink is not created during a ceph-disk prepare run.

Environment:

- Scientific Linux 6.6
- ceph-disk from master branch
- /dev/sdo is a new empty spinning disk (for the OSD)
- /dev/sdc is an SSD with 5 journal partitions
- /dev/sdc1 is not currently used by any OSD

To reproduce:

- ceph-disk --verbose prepare /dev/sdo /dev/sdc1

Expected result:

- sdo becomes and OSD with a sdc1 as the journal. The /var/lib/ceph/osd/ceph-X/journal should be soft-linked to /dev/disk/by-partuuid/<uuid of sdc1> which is a softlink to /dev/sdc1

Actual result:

- /var/lib/ceph/osd/ceph-X/journal is softlinked to /dev/disk/by-partuuid/<uuid of sdc1>, but /dev/disk/by-partuuid/<uuid of sdc1> is a plain empty file, *not* a softlink to /dev/sdc1

Explanation:

- In function prepare_journal_dev sgdisk is called to change the partition guid, then partx -a is called to reload the partition table, the udevadm settle is called to let udev finish handling the new ptable. It is expected that either sgdisk or partx triggers udev to add the new /dev/disk/by-partuuid/ symlink to /dev/sdc1, but in practise (with a busy server) the new symlink is not created. By "busy", we mean that /dev/sdc is seeing around 100 writes / second.
- Since the by-partuuid symlink doesn't exist, later in ceph-disk when the symlink from /var/lib/ceph/osd/ceph-X/journal to /dev/disk/by-partuuid/<journal_uuid> is made, this results in an empty file being created at the link target, and afterwords the OSD cannot start.

Solutions:

- We have found that by retriggering the udev block subsystem the symlink is always created. See the patch here: https://github.com/ceph/ceph/pull/2955
- Another possible solution would be to *not* change the partition guid when re-using a journal partition. The previous /dev/disk/by-partuuid/ link would already exist and could be used by the new OSD.

Related issues:

Copied to Ceph - Backport #12418: ceph-disk: sometimes the journal symlink is...	**Resolved**	**11/20/2014**

Associated revisions

Revision 29eb1350 - 12/13/2014 02:01 PM - Dan van der Ster

ceph-disk: don't change the journal partition uuid

We observe that the new /dev/disk/by-partuuid/<journal_uuid>
symlink is not always created by udev when reusing a journal
partition. Fix by not changing the uuid of a journal partition
in this case -- instead we can reuse the existing uuid (and
journal_symlink) instead. We also now assert that the symlink
exists before further preparing the OSD.

Fixes: #10146
Signed-off-by: Dan van der Ster <daniel.vanderster@cern.ch>
Tested-by: Dan van der Ster <daniel.vanderster@cern.ch>

Revision e0f052a1 - 12/13/2014 02:01 PM - Dan van der Ster

ceph-disk: test re-using an existing journal partition

Add a ceph-disk test to first setup an OSD with a separate journal
block device, then tear down the OSD (simulating a failure) and create
a new OSD which re-uses the same journal device.

Add create_dev / destroy_dev helpers that encapsulate the operations
that ensure the partition table is up to date in the kernel and the
symlinks are created as expected. In particular it makes sure the kernel
is aware that the partition table of a newly created device is
empty. If the device previously existed and the kernel was not informed
of the latest partition table updates via partprobe / partx, it may
have cached an old partition table which can create all sorts of
unexpected behaviors such as a failure to create the by-partuuid
symbolic links as described in http://tracker.ceph.com/issues/10146
Refs: #10146

Signed-off-by: Dan van der Ster <daniel.vanderster@cern.ch>
Signed-off-by: Loic Dachary <ldachary@redhat.com>

Revision 04b2a878 - 08/11/2015 08:19 AM - Dan van der Ster

ceph-disk: don't change the journal partition uuid

We observe that the new /dev/disk/by-partuuid/<journal_uuid>
symlink is not always created by udev when reusing a journal
partition. Fix by not changing the uuid of a journal partition
in this case -- instead we can reuse the existing uuid (and
journal_symlink) instead. We also now assert that the symlink
exists before further preparing the OSD.

Fixes: #10146
Signed-off-by: Dan van der Ster <daniel.vanderster@cern.ch>
Tested-by: Dan van der Ster <daniel.vanderster@cern.ch>
(cherry picked from commit 29eb1350b4acaeabfe1d2b19efedbce22641d8cc)

History

#1 - 11/20/2014 02:15 AM - Loïc Dachary

- Status changed from New to In Progress

- Assignee set to Dan van der Ster

I like the idea of not changing the uuid

#2 - 11/20/2014 06:50 AM - Dan van der Ster

I've pushed the alternative fix in the same pull req.

#3 - 12/02/2014 01:10 PM - Sage Weil

- Status changed from In Progress to Resolved

#4 - 12/02/2014 01:11 PM - Loïc Dachary

- Status changed from Resolved to In Progress

Still open, needs tests.

#5 - 12/13/2014 01:48 AM - Loïc Dachary

I'm able to reproduce that frequently by running

```
sudo test/ceph-disk.sh test_activate_journal_dev
```

on my laptop at the moment. I'm taking that opportunity to find the cause.

#6 - 12/13/2014 02:56 AM - Loïc Dachary

I think what happens is the following sequence

- partition 1 is created
- partprobe called so the kernel notices (the symlink shows)
- partition table is zapped and symlink removed
- partproble is **not** called
- partition 1 is created
- partprobe called but the kernel thinks partition already exists and does not trigger an udev event that does not create the symlink

the only way to reset the idea the kernel has about a given device is to zap the partition table + partprobe. I think symlinks are created reliably and it does not depend on the machine load.

#7 - 12/15/2014 12:55 AM - Dan van der Ster

Hi Loic,
In this case, ceph-disk zap doesn't apply. The use-case is that you have say 4-5 partitions on a shared journal SSD, and you want to re-use only one of those partitions to become the journal for a new OSD. So we don't (and mustn't!) call ceph-disk zap on the SSD device in this case.

Instead we tried changing the guid of the journal partition, but that doesn't trigger udev reliably. So the only reliable method is *not* to change the guid, as in the current pull req.
Cheers, Dan

#8 - 12/15/2014 01:11 AM - Loïc Dachary

Dan van der Ster wrote:

> Instead we tried changing the guid of the journal partition, but that doesn't trigger udev reliably. So the only reliable method is *not* to change the guid, as in the current pull req.

I'm under the impression (although I've never actually tried to prove it) that the udev event will never be called if the guid is modified, even though it makes the by-partuuid symlink obsolete. Not changing the guid (which is what you implemented at https://github.com/ceph/ceph/commit/29eb1350b4acaeabfe1d2b19efedbce22641d8cc) works around the problem.

This should probably be a bug report against udev ?

#9 - 12/15/2014 01:23 AM - Dan van der Ster

Loic Dachary wrote:

> I'm under the impression (although I've never actually tried to prove it) that the udev event will never be called if the guid is modified, even though it makes the by-partuuid symlink obsolete.

On our test cluster changing the guid *did* trigger udev and make the correct by-partuuid link (and also left behind the old link -- no big deal). On our prod cluster changing the guid *did not* trigger udev. Both are CentOS 6.6. The only difference is the activity on the journal devs.

#10 - 12/15/2014 01:36 AM - Loïc Dachary

Ok. Have you ever seen a problem under load where udev would fail to notice the creation / removal of a partition although partprobe / partx is called consistently (i.e. after each creation / removal) ?

#11 - 12/15/2014 01:47 AM - Dan van der Ster

For OSD devices (one dev -- one OSD), I haven't observed any problems, regardless of load. (In this case, the OSD process is not running, so no processes have the device open when the partition is removed or created).

For journal devices, it really depends. Removing 1 out of 5 journal partitions (when the other 4 are still used by active OSDs) is not really doable, in my experience with CentOS 6. I never found the combination of partprobe / partx that made the OSD realize the new ptable. The only reliable method to remove/recreate partitions on a shared journal dev was to *stop* all OSDs using that dev, then adjust the ptable, then restart the OSDs.

#12 - 12/15/2014 02:17 AM - Loïc Dachary

Thanks for explaining, that makes me slightly less worried about the reliability of the partition/udev notification couple in the most common case :-)

#13 - 01/13/2015 01:29 PM - Loïc Dachary

- Status changed from In Progress to Resolved

https://github.com/ceph/ceph/pull/3172

#14 - 07/21/2015 12:11 PM - Loïc Dachary

- Status changed from Resolved to Pending Backport

- Backport set to firefly

- Regression set to No

#15 - 10/20/2015 11:05 AM - Loïc Dachary

- Status changed from Pending Backport to Resolved