RADOS - Bug #35542

Backfill and recovery should validate all checksums

09/04/2018 06:27 PM - Greg Farnum

Status:	Won't Fix	**Start date:**	09/04/2018
Priority:	High	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Scrub/Repair	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	v12.2.5
Backport:		**ceph-qa-suite:**	
Regression:	No	**Component(RADOS):**	
Severity:	3 - minor	**Pull request ID:**	

Description

From the thread "Copying without crc check when peering may lack reliability" on ceph-devel, it appears that backfill does not validate a read object against the checksums that may be stored in hobject. We should not silently transfer data which will be detected as bad on read — we should detect it as bad and behave rationally in response! (Crash out on primary, trigger recovery, something.)

```
Next we put an object into the pool.

-> # cat txt
123
-> # rados -p test put test_copy txt
-> # rados -p test get test_copy -
123

Then we make OSD.0 down, and change its data of object test_copy.

-> # ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0
test_copy get-bytes
123
-> # ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-0
test_copy set-bytes 120txt

Next we start OSD.0 and do data migration.

-> # ceph osd pool set test crush_rule root1_rule

Finally we try to get the object by rados and ceph-objectstore-tool

-> # rados -p test get test_copy -
error getting test/test_copy: (5) Input/output error
-> # ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-3
test_copy get-bytes
120
-> # ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-4
test_copy get-bytes
120
-> # ceph-objectstore-tool --data-path /var/lib/ceph/osd/ceph-5
test_copy get-bytes
120

The data of test_copy on OSD.3 OSD.4 OSD.5 is from OSD.0 which has the
silent data corruption.
```

History

#1 - 09/04/2018 06:27 PM - Greg Farnum

Oh, this may just be 12.2.5 being broken? In which case we can close.

#2 - 09/04/2018 06:40 PM - Greg Farnum

Nope, 12.2.6 was the one that didn't handle checksums properly. So this looks like a real issue, although I think we also tried to move away from the double-checksumming...?

#3 - 09/12/2018 03:34 PM - Sage Weil

I'm unclear what checksum is not being checked. There is only **sometimes** a full object checksum that we can validate against--unclear from this whether one was set. The bluestore checksum is always checked, but it won't be "corrupt" here because ceph-objectstore-tool writes via bluestore and the crc at that layer will be updated to remain consistent.

#4 - 09/12/2018 08:07 PM - Greg Farnum

Sage Weil wrote:

> I'm unclear what checksum is not being checked. There is only **sometimes** a full object checksum that we can validate against--unclear from this whether one was set. The bluestore checksum is always checked, but it won't be "corrupt" here because ceph-objectstore-tool writes via bluestore and the crc at that layer will be updated to remain consistent.

Sure, but if on a read we're getting EIO then there must in fact be the full-object checksum. And since it was just recovered on backfill, that means the read is being more careful than the backfill op was. It's easy for me as a developer to see how that happened, but from a user perspective it's quite confusing and it doesn't fit our general "as-safe-as-possible" strategy.

#5 - 08/20/2019 11:01 PM - David Zafman

- Status changed from New to Won't Fix

Bluestore makes this unnecessary and it is only possible on a pull of the complete object.