RADOS - Bug #24875

OSD: still returning EIO instead of recovering objects on checksum errors

07/11/2018 08:25 PM - Greg Farnum

Status:	Resolved	**Start date:**	07/11/2018
Priority:	High	**Due date:**	
Assignee:	David Zafman	**% Done:**	0%
Category:	Scrub/Repair	**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:	mimic, luminous	**ceph-qa-suite:**	
Regression:	No	**Component(RADOS):**	OSD
Severity:	3 - minor	**Pull request ID:**	

Description

A report came in on the mailing list of an MDS journal which couldn't be read and was throwing errors:

```
2018-07-11 15:49:14.913771 7efbee672700 -1 log_channel(cluster) log [ERR] : 10.14 full-object read
 crc 0x976aefc5 != expected 0x9ef2b41b on 10:292cf221:::200.00000000:head
```

And indeed, when you search for that log message it pops up in PrimaryLogPG::do_read() and do_sparse_read() (and also struct FillInVerifyExtent). When it pops up, the function returns -EIO, and do_osd_ops() (which is the only caller) turns that into a direct client return.
There's a comment "try repair later" which makes me think the author expected the EIO to get turned into a read-repair, but tracing back through git history there's no indication of any work done to enable that in this path.

Related issues:

Related to RADOS - Bug #25084: Attempt to read object that can't be repaired ...	**Resolved**	**07/24/2018**
Copied to RADOS - Backport #25226: mimic: OSD: still returning EIO instead of...	**Resolved**	
Copied to RADOS - Backport #25227: luminous: OSD: still returning EIO instead...	**Resolved**	

History

#1 - 07/11/2018 09:01 PM - David Zafman

The do_sparse_read() path doesn't attempt to repair a checksum error. Could that be the real issue?

The do_read() path looks fine since it calls rep_repair_primary_object() whether the EIO came from the disk or the crc check.

```
      if (oi.data_digest != crc) {
        osd->clog->error() << info.pgid << std::hex
                           << " full-object read crc 0x" << crc
                           << " != expected 0x" << oi.data_digest
                           << std::dec << " on " << soid;
        r = -EIO; // try repair later
      }
    }
    if (r == -EIO) {
      r = rep_repair_primary_object(soid, ctx->op);
    }
```

#2 - 07/11/2018 09:38 PM - Greg Farnum

Ah, the error was reported on luminous, which doesn't do the repair, and I guess I missed it on master. Sorry for the mis-diagnosis.

(Looks like the MDS doesn't use sparse-read, but we should definitely still fix that path too!)

#3 - 07/11/2018 11:06 PM - Josh Durgin

- Priority changed from Normal to High

#4 - 07/12/2018 02:05 PM - Dan van der Ster

Is this the relevant fix? https://github.com/ceph/ceph/commit/4667280f8afe6cd68dfffea61d7530581f3dd0eb

Alessandro's OSDs are bluestore, and he doesn't get any bluestore block checksum errors. So the crc can be wrong when the bluestore data is correct?

Will the above patch correct this type of crc error?

Also, we deep-scrubbed the PG and it didn't find any inconsistent objects.

#5 - 07/13/2018 03:00 PM - Dan van der Ster

FTR, this crc issue is probably due to an incomplete backport to 12.2.6 of the skip_digest changes for bluestore:

```
[12:56:59]  <dvanders> regarding the 12.2.6 cephfs crc errors, could it be `b519a0b1c1 osd/PrimaryLogPG: do no
t generate data digest for BlueStore by default`  or one of the other omap/data_digest changes that landed in
12.2.6 ?
[12:59:29]  <dvanders> Seems similar to https://tracker.ceph.com/issues/23871 ... which was fixed in mimic but
 not luminous: `fe5038c7f9 osd/PrimaryLogPG: clear data digest on WRITEFULL if skip_data_digest`
[14:10:16]  <sage>    dvanders: yeah does seem similar, but i'm not sure why it would manifest during a .5 to
.6 upgrade.  looking...
[14:10:37]  <sage>    dvanders: it's definitely bluestore-only?
[14:11:20]  <dvanders>    i didn't try filestore, but all the clusters i've seen were bluestore
[14:14:56]  <sage>    ah, it's becaues teh other skip_digest handlng code was just backporting/changed
[14:14:59]  <sage>    backported/changed
```

This issue is related to https://tracker.ceph.com/issues/23871

#6 - 08/01/2018 10:33 PM - David Zafman

- Related to Bug #25084: Attempt to read object that can't be repaired loops forever added

#7 - 08/01/2018 10:58 PM - David Zafman

- Status changed from New to Verified

- Assignee set to David Zafman

#8 - 08/01/2018 10:58 PM - David Zafman

- Backport set to mimic, luminous

#9 - 08/01/2018 11:32 PM - David Zafman

- Copied to Backport #25226: mimic: OSD: still returning EIO instead of recovering objects on checksum errors added

#10 - 08/01/2018 11:37 PM - David Zafman

- Copied to Backport #25227: luminous: OSD: still returning EIO instead of recovering objects on checksum errors added

#11 - 08/02/2018 03:59 AM - Nathan Cutler

master PR: https://github.com/ceph/ceph/pull/23377

#12 - 08/03/2018 10:18 PM - David Zafman

- Status changed from Verified to In Progress

#13 - 08/06/2018 02:51 PM - Kefu Chai

- Status changed from In Progress to Pending Backport

#14 - 08/22/2018 09:00 PM - Nathan Cutler

- Status changed from Pending Backport to Resolved