CephFS - Bug #51589

mds: crash when journaling during replay

07/08/2021 07:44 AM - □□ □

Status:	Resolved	**% Done:**	0%
Priority:	Normal		
Assignee:	Venky Shankar		
Category:			
Target version:	v17.0.0		
Source:	Community (user)	**ceph-qa-suite:**	
Tags:	backport_processed	**Component(FS):**	MDS
Backport:	pacific,octopus	**Labels (FS):**	crash
Regression:	No	**Pull request ID:**	
Severity:	3 - minor	**Crash signature (v1):**	
Reviewed:		**Crash signature (v2):**	
Affected Versions:	v14.2.20		

Description

MDS version: ceph version 14.2.20 (36274af6eb7f2a5055f2d53ad448f2694e9046a0) nautilus (stable)

Using 200 clients, mds crashed after writing for many days.

But I don’t know what caused the mds to crash.

```
[twj@xxxxxxxxx-MN-001.sn.cn ~]$ sudo ceph fs status
cephfs - 200 clients
======
+------+----------------+-----------------------+----------+-------+-------+
| Rank |     State      |          MDS          | Activity |  dns  |  inos |
+------+----------------+-----------------------+----------+-------+-------+
|  0   |    resolve     | xxxxxxxxxxMN-002.sn.cn |          |    0  |    3  |
|  1   | resolve(laggy) | xxxxxxxxxxMN-003.sn.cn |          |    0  |    0  |
+------+----------------+-----------------------+----------+-------+-------+
+---------------------+----------+-------+-------+
|         Pool        |   type   |  used | avail |
+---------------------+----------+-------+-------+
| cephfs.metadata.pool | metadata | 70.5G |  793G |
|  cephfs.data.pool1   |   data   |  183T | 1115T |
|  cephfs.data.pool2   |   data   |  299T | 1042T |
+---------------------+----------+-------+-------+
+-------------+
| Standby MDS |
+-------------+
+-------------+
MDS version: ceph version 14.2.20 (36274af6eb7f2a5055f2d53ad448f2694e9046a0) nautilus (stable)
```

All mds crashed for this reason:

```
    -1> 2021-07-08 15:14:13.283 7f3804255700 -1 /builddir/build/BUILD/ceph-14.2.20/src/mds/MDLog.cc: In function 'void MDLog::_submit_entry(LogEvent*, MDSLogContextBase*)' thread 7f3804255700 time 2021-07-08 15:14:13.283719
/builddir/build/BUILD/ceph-14.2.20/src/mds/MDLog.cc: 288: FAILED ceph_assert(!segments.empty())

 ceph version 14.2.20 (36274af6eb7f2a5055f2d53ad448f2694e9046a0) nautilus (stable)
 1: (ceph::__ceph_assert_fail(char const*, char const*, int, char const*)+0x14a) [0x7f380d72cfe7]
 2: (()+0x25d1af) [0x7f380d72d1af]
 3: (MDLog::_submit_entry(LogEvent*, MDSLogContextBase*)+0x599) [0x557471ec5959]
 4: (Server::journal_close_session(Session*, int, Context*)+0x9ed) [0x557471c7e02d]
 5: (Server::kill_session(Session*, Context*)+0x234) [0x557471c81914]
 6: (Server::apply_blacklist(std::set<entity_addr_t, std::less<entity_addr_t>, std::allocator<enti
ty_addr_t> > const&)+0x14d) [0x557471c8449d]
 7: (MDSRank::reconnect_start()+0xcf) [0x557471c49c5f]
 8: (MDSRankDispatcher::handle_mds_map(boost::intrusive_ptr<MMDSMap const> const&, MDSMap const&)+
0x1c29) [0x557471c57979]
 9: (MDSDaemon::handle_mds_map(boost::intrusive_ptr<MMDSMap const> const&)+0xa9b) [0x557471c3091b]
 10: (MDSDaemon::handle_core_message(boost::intrusive_ptr<Message const> const&)+0xed) [0x557471c3
216d]
 11: (MDSDaemon::ms_dispatch2(boost::intrusive_ptr<Message> const&)+0xc3) [0x557471c32983]
 12: (DispatchQueue::entry()+0x1699) [0x7f380d952b79]
 13: (DispatchQueue::DispatchThread::entry()+0xd) [0x7f380da008ed]
 14: (()+0x7ea5) [0x7f380b5eeea5]
 15: (clone()+0x6d) [0x7f380a29e96d]
```

Related issues:

Copied to CephFS - Backport #52952: pacific: mds: crash when journaling durin...	**Resolved**
Copied to CephFS - Backport #52953: octopus: mds: crash when journaling durin...	**Resolved**

History

#1 - 07/08/2021 01:29 PM - Igor Fedotov

- Project changed from bluestore to CephFS

#2 - 07/08/2021 02:20 PM - Patrick Donnelly

- Description updated

- ceph-qa-suite deleted (fs)

- Component(FS) MDS added

- Labels (FS) crash added

#3 - 07/12/2021 01:42 PM - Patrick Donnelly

- Status changed from New to Triaged

- Assignee set to Venky Shankar

- Target version set to v17.0.0

- Source set to Community (user)

#4 - 07/28/2021 04:40 AM - Venky Shankar

- Status changed from Triaged to In Progress

#5 - 08/11/2021 01:20 PM - Venky Shankar

If I'm reading this correctly, it looks like an log entry can be submitted when there are no log segments available for logging (in MDLog). I'm going to see if this exists in master (or pacific) in case we missed backporting the fix.

#6 - 08/18/2021 09:07 AM - Venky Shankar

Normally, MDLog does not expire the latest log segment during trimming. The exception to this is when then log segments are not meant to be capped (via mdlog->cap()). This happens when the MDS is stopping via MDCache which calls mdlolg->cap). However, client requests are not processes at this time. So, its a bit hard to tell if this is what caused the crash (especially without the logs).

And, BTW, I do not see a missing backport which means this can be still hit in master and other releases.

#7 - 08/24/2021 12:23 PM - Venky Shankar

I'm unable to reproduce the crash. Do you have the MDS logs at the time of crash? As you mentioned you do not have the steps that caused the MDS to crash, any pointers to what operations were performed would help.

#8 - 09/29/2021 12:56 PM - Venky Shankar

I was able to reproduce this in master branch. The crash happens when a standby mds takes over as active and there are blocklisted clients. During replay state, the (new) active mds tries to journal the blocklisted client information, however, in this state (resolve) the journal segments are still not available to journaling, leading to an crash. One minor difference is that the backtrace mentioned in this tracker description hints at the assert happening during reconnect state which I wasn't able to hit (happens in replay state rather than reconnect). However, this tracker mentions that the crash is in nautilus (while I was able to hit it in current master).

Patrick suggested that we could defer journaling blocklisted clients in reconnect state where journal segments are available, but again, the backtrace in this tracker shows that it can happen in reconnect state. Maybe that happens in nautilus, I'm not 100% sure about that.

I'll update the tracker when I have a fix in mind.

#9 - 09/30/2021 08:43 AM - Venky Shankar

Venky Shankar wrote:

> ...
> ...
> Patrick suggested that we could defer journaling blocklisted clients in reconnect state where journal segments are available, but again, the backtrace in this tracker shows that it can happen in reconnect state. Maybe that happens in nautilus, I'm not 100% sure about that.

Another approach could be to ignore journaling blocklisted clients in `resolve` state (since this happens when handling an OSD Map in MDSRank::handle_osd_map). Subsequently, the MDS will switch to `reconnect` state where the blocklisted clients get journaled again.

#10 - 10/14/2021 12:46 PM - Venky Shankar

Partially fixed with https://github.com/ceph/ceph/pull/43382

#11 - 10/14/2021 12:46 PM - Venky Shankar

- Backport set to pacific,octopus

#12 - 10/15/2021 03:12 PM - Patrick Donnelly

- Subject changed from mds crash to mds: crash when journaling during replay

- Status changed from In Progress to Pending Backport

#13 - 10/15/2021 03:15 PM - Backport Bot

- Copied to Backport #52952: pacific: mds: crash when journaling during replay added

#14 - 10/15/2021 03:15 PM - Backport Bot

- Copied to Backport #52953: octopus: mds: crash when journaling during replay added

#15 - 08/08/2022 04:32 PM - Backport Bot

- Tags set to backport_processed

#16 - 09/06/2022 05:36 AM - Venky Shankar

- Status changed from Pending Backport to Resolved