

Messengers - Bug #10080

Pipe::connect() cause osd crash when osd reconnect to its peer

11/12/2014 03:40 AM - Wenjun Huang

Status: Resolved	% Done: 0%
Priority: Normal	
Assignee: Greg Farnum	
Category:	
Target version:	
Source: Community (user)	Affected Versions:
Tags:	ceph-qa-suite:
Backport: giant, firefly	Pull request ID:
Regression: No	Crash signature (v1):
Severity: 3 - minor	Crash signature (v2):
Reviewed:	

Description

When our cluster load is heavy, the osd sometimes crashes. The critical log is as below:

```
278> 2014-08-20 11:04:28.609192 7f89636c8700 10 osd.11 783 OSD::ms_get_authorizer type=osd
-277> 2014-08-20 11:04:28.609783 7f89636c8700 2- 10.193.207.117:6816/44281 >> 10.193.207.125:6804/2022817
pipe(0x7ef2280 sd=105 :42657 s=1 pgs=236754 cs=4 l=0 c=0x44318c0). got newly_acked_seq 546 vs out_seq 0
276> 2014-08-20 11:04:28.609810 7f89636c8700 2- 10.193.207.117:6816/44281 >> 10.193.207.125:6804/2022817
pipe(0x7ef2280 sd=105 :42657 s=1 pgs=236754 cs=4 l=0 c=0x44318c0). discarding previously sent 1 osd_map(727..755 src has
1..755) v3
275> 2014-08-20 11:04:28.609859 7f89636c8700 2- 10.193.207.117:6816/44281 >> 10.193.207.125:6804/2022817
pipe(0x7ef2280 sd=105 :42657 s=1 pgs=236754 cs=4 l=0 c=0x44318c0). discarding previously sent 2 pg_notify(1.2b(22),2.2c(23)
epoch 755) v5

2014-08-20 11:04:28.608141 7f89629bb700 0 -- 10.193.207.117:6816/44281 >> 10.193.207.125:6804/2022817 pipe(0x7ef2280
sd=134 :6816 s=2 pgs=236754 cs=3 l=0 c=0x44318c0).fault, initiating reconnect
2014-08-20 11:04:28.609192 7f89636c8700 10 osd.11 783 OSD::ms_get_authorizer type=osd
2014-08-20 11:04:28.666895 7f89636c8700 -1 msg/Pipe.cc: In function 'int Pipe::connect()' thread 7f89636c8700 time 2014-08-20
11:04:28.618536 msg/Pipe.cc: 1080: FAILED assert(m)
```

Looking into the log, we can see, the out_seq is 0. As our cluster has enabled the cephx authorization, from the source code, I am informed that the out_seq is initialized by a random number. So there should be some bugs in the source code.

We face the crash issue, almost every time our cluster load is heavy. So, I think it is a critical bug for ceph.

Associated revisions

Revision 8cd1fdd7 - 12/03/2014 05:33 AM - Greg Farnum

SimpleMessenger: allow RESETSESSION whenever we forget an endpoint

In the past (e229f8451d37913225c49481b2ce2896ca6788a2) we decided to disable reset of lossless Pipes, because lossless peers resetting caused trouble and they can't forget about each other. But they actually can: if mark_down() is called.

I can't figure out how else we could forget about a remote endpoint, so I think it's okay if we tell them we reset in order to clean up state. That's desirable so that we don't get into strange situations with out-of-whack counters.

Fixes: #10080

Backport: giant, firefly, dumpling

Signed-off-by: Greg Farnum <gfarnum@redhat.com>

Revision 7ed92f7d - 03/09/2015 08:07 PM - Greg Farnum

SimpleMessenger: allow RESETSESSION whenever we forget an endpoint

In the past (e229f8451d37913225c49481b2ce2896ca6788a2) we decided to disable reset of lossless Pipes, because lossless peers resetting caused trouble and they can't forget about each other. But they actually can: if mark_down() is called.

I can't figure out how else we could forget about a remote endpoint, so I think it's okay if we tell them we reset in order to clean up state. That's desirable so that we don't get into strange situations with out-of-whack counters.

Fixes: #10080

Backport: giant, firefly, dumpling

Signed-off-by: Greg Farnum <gfarnum@redhat.com>
(cherry picked from commit 8cd1fdd7a778eb84cb4d7161f73bc621cc394261)

Revision 1ccf5835 - 03/19/2015 09:04 AM - Greg Farnum

SimpleMessenger: allow RESETSESSION whenever we forget an endpoint

In the past (e229f8451d37913225c49481b2ce2896ca6788a2) we decided to disable reset of lossless Pipes, because lossless peers resetting caused trouble and they can't forget about each other. But they actually can: if mark_down() is called.

I can't figure out how else we could forget about a remote endpoint, so I think it's okay if we tell them we reset in order to clean up state. That's desirable so that we don't get into strange situations with out-of-whack counters.

Fixes: #10080

Backport: giant, firefly, dumpling

Signed-off-by: Greg Farnum <gfarnum@redhat.com>
(cherry picked from commit 8cd1fdd7a778eb84cb4d7161f73bc621cc394261)

History

#1 - 11/12/2014 04:26 AM - Wenjun Huang

And the peer OSD's log is as below:

```
2014-08-20 10:49:28.462222 7fae57f29700 0 -- 10.193.207.125:6804/2022817 >> 10.193.207.117:6816/44281 pipe(0x1dfcea00 sd=74 :53319 s=2 pgs=70251 cs=3 l=0 c=0x1cdcc420).reader got old message 264 <= 546 0x1d368fc0 osd_map(783..783 src has 1..783) v3, discarding
2014-08-20 10:49:28.462363 7fae57f29700 0 -- 10.193.207.125:6804/2022817 >> 10.193.207.117:6816/44281 pipe(0x1dfcea00 sd=74 :53319 s=2 pgs=70251 cs=3 l=0 c=0x1cdcc420).reader got old message 265 <= 546 0x1d7c5340 pg_notify(1.2b(22) epoch 783) v5, discarding
...
2014-08-20 10:49:28.503780 7fae57f29700 0 -- 10.193.207.125:6804/2022817 >> 10.193.207.117:6816/44281 pipe(0x1dfcea00 sd=74 :53319 s=2 pgs=70251 cs=3 l=0 c=0x1cdcc420).reader got old message 272 <= 546 0x2082afc0 pg_notify(7.2a6s2(40) epoch 783) v5, discarding

2014-08-20 11:04:28.608540 7fae57f29700 0 -- 10.193.207.125:6804/2022817 >> 10.193.207.117:6816/44281 pipe(0x1dfcea00 sd=74 :53319 s=2 pgs=70251 cs=3 l=0 c=0x1cdcc420).fault with nothing to send, going to standby
2014-08-20 11:04:28.609780 7fae42b4d700 10 osd.46 783 new session 0x1d7ccee0 con=0x208586e0 addr=10.193.207.117:6816/44281
2014-08-20 11:04:28.609848 7fae42b4d700 10 osd.46 783 session 0x1d7ccee0 osd.11 has caps osdcap[grant(*)] 'allow *'
2014-08-20 11:04:28.609860 7fae42b4d700 0 -- 10.193.207.125:6804/2022817 >> 10.193.207.117:6816/44281 pipe(0x1ff1c780 sd=148 :6804 s=0 pgs=0 cs=0 l=0 c=0x208586e0).accept connect_seq 4 vs existing 3 state standby
2014-08-20 11:04:28.609905 7fae76510700 1 osd.46 783 ms_handle_reset con 0x208586e0 session 0x1d7ccee0
2014-08-20 11:04:29.360535 7fae8a54f700 5 osd.46 783 tick
2014-08-20 11:04:29.360574 7fae8a54f700 10 osd.46 783 do_waiters -- start
2014-08-20 11:04:29.360581 7fae8a54f700 10 osd.46 783 do_waiters -- finish
2014-08-20 11:04:29.585448 7fae7510e700 10 osd.46 783 heartbeat_reset failed hb con 0x1e998580 for osd.11, reopening
2014-08-20 11:04:29.585658 7fae42b4d700 0 -- 10.193.207.125:6804/2022817 >> 10.193.207.117:6816/44281 pipe(0x1ff1c780 sd=148 :6804 s=3 pgs=76640 cs=5 l=0 c=0x1cdcc420).fault with nothing to send, going to standby
```

#2 - 11/12/2014 10:47 AM - Greg Farnum

- Project changed from CephFS to Ceph
- Category set to msg

What version are you running? This looks like one of a couple of bugs that have been resolved in the latest point releases.

#3 - 11/12/2014 06:52 PM - Wenjun Huang

Greg Farnum wrote:

What version are you running? This looks like one of a couple of bugs that have been resolved in the latest point releases.

We use the version 0.80.4, I have walked through the release note since that version, it seems that there is not a related bug fix about the issue.

Could you have confirm? And if there is any clue, feel free to let me know.

#4 - 11/24/2014 12:59 AM - Guang Yang

I am wondering if the following race occurred:

Let us assume A and B are two OSDs having the connection (pipe) between each other.

1. B issued a re-connection for whatever reason, and at the same time, A marked down and destroyed the Pipe with B.
2. Let us assume B having: cs = 100, in_seq = 500
3. The connection is established with cs = 101
4. For whatever reason, A came across a failure during read and issued a new connection request with cs = 102, currently A has out_seq = 0 (as it is a brand new Pipe) and cs = 102.
5. B accepted the connection request and responded back in_seq = 500 (already wrong here)
6. A got the in_seq and comparing with its internal out_seq and out_q, crashed with assertion failure.

If this is the case, it seems one step was missed during seq negotiation, that is, when B tried to do a new connection and detected that A has a reset, it should reset its in_seq as well?

Thanks,
Guang

#5 - 11/26/2014 11:34 PM - Guang Yang

Add some peer's logs to prove the two-ways connect:

```
...
2014-10-28 19:58:50.289534 7f20de7f6700 0 -- 10.214.143.136:6801/3028784 >> 10.214.143.72:6823/9030193 pipe(0
x2eab6a00 sd=23 :51260 s=2 pgs=539645 cs=535 l=0 c=0x1fcb9a20).fault, initiating reconnect
2014-10-28 19:58:50.293257 7f20de7f6700 0 -- 10.214.143.136:6801/3028784 >> 10.214.143.72:6823/9030193 pipe(0
x2eab6a00 sd=23 :51283 s=2 pgs=539646 cs=537 l=0 c=0x1fcb9a20).fault, initiating reconnect
2014-10-28 19:58:50.296343 7f20de7f6700 0 -- 10.214.143.136:6801/3028784 >> 10.214.143.72:6823/9030193 pipe(0
x2eab6a00 sd=23 :51305 s=2 pgs=539647 cs=539 l=0 c=0x1fcb9a20).fault, initiating reconnect
2014-10-28 20:25:20.888594 7f2082b2e700 0 -- 10.214.143.136:0/4028784 >> 10.214.143.72:6823/9030193 pipe(0x3a
d42000 sd=52 :33518 s=2 pgs=545030 cs=1 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:20.935009 7f2082b2e700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=52 :33527 s=2 pgs=545031 cs=3 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:20.936264 7f2082b2e700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=24 :33531 s=2 pgs=545032 cs=5 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:20.937338 7f2082b2e700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=24 :33535 s=2 pgs=545033 cs=7 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:20.938806 7f2082b2e700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=24 :33539 s=2 pgs=545034 cs=9 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:20.940500 7f2082b2e700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=24 :33545 s=2 pgs=545035 cs=11 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:20.941595 7f2082b2e700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=24 :33548 s=2 pgs=545036 cs=13 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:20.942752 7f2082b2e700 0 -- 10.214.143.136:6831/5028784 >> initiating reconnect
014-10-28 20:25:25.632142 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0x
3ad42000 sd=129 :52237 s=2 pgs=547312 cs=4563 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.635890 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=129 :52289 s=2 pgs=547313 cs=4565 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.638601 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=129 :52357 s=2 pgs=547314 cs=4567 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.643267 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=129 :52402 s=2 pgs=547315 cs=4569 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.645738 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
```

```
x3ad42000 sd=129 :52465 s=2 pgs=547316 cs=4571 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.649585 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=129 :52483 s=2 pgs=547317 cs=4573 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.651460 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=129 :52523 s=2 pgs=547318 cs=4575 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.653400 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=129 :52542 s=2 pgs=547319 cs=4577 l=0 c=0xlcccfce0).fault, initiating reconnect
2014-10-28 20:25:25.673181 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 1 <= 2454 0x362d2d80 o
sd_map(15398..15402 src has 1..15402) v3, discarding
2014-10-28 20:25:25.680228 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 2 <= 2454 0x21896780 p
g_query(4.323 epoch 15402) v3, discarding
2014-10-28 20:25:25.930323 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 3 <= 2454 0x21896780 p
g_query(4.323 epoch 15402) v3, discarding
2014-10-28 20:25:45.706962 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 4 <= 2454 0x13862000 o
sd_map(15416..15416 src has 1..15416) v3, discarding
2014-10-28 20:27:40.807687 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 5 <= 2454 0x16169680 o
sd_map(15452..15452 src has 1..15452) v3, discarding
2014-10-28 20:27:46.113404 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 6 <= 2454 0x16169680 o
sd_map(15453..15454 src has 1..15454) v3, discarding
2014-10-28 20:27:47.822336 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 7 <= 2454 0x16169680 o
sd_map(15455..15456 src has 1..15456) v3, discarding
2014-10-28 20:27:51.930894 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 8 <= 2454 0x16169680 o
sd_map(15457..15459 src has 1..15459) v3, discarding
2014-10-28 20:27:52.950391 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 9 <= 2454 0x16169680 o
sd_map(15460..15460 src has 1..15460) v3, discarding
2014-10-28 20:27:55.764524 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 10 <= 2454 0x16169680
osd_map(15461..15461 src has 1..15461) v3, discarding
2014-10-28 20:27:58.070885 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).reader got old message 11 <= 2454 0x16169680
osd_map(15462..15463 src has 1..15463) v3, discarding
2014-10-28 20:46:34.333831 7f20c3242700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x3ad42000 sd=409 :52556 s=2 pgs=547320 cs=4579 l=0 c=0xlcccfce0).fault with nothing to send, going to standby
2014-10-28 20:46:34.334979 7f207eff3700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x32285900 sd=258 :6831 s=0 pgs=0 cs=0 l=0 c=0x2eec58c0).accept connect_seq 4580 vs existing 4579 state standby
2014-10-28 20:46:37.362160 7f207eff3700 0 -- 10.214.143.136:6831/5028784 >> 10.214.143.72:6823/9030193 pipe(0
x32285900 sd=258 :6831 s=3 pgs=547352 cs=4581 l=0 c=0xlcccfce0).fault with nothing to send, going to standby
```

At some point B rebinds and connect to A, even though the connection has been established, the connection/pipe state is wrong (A's out_seq/out_q and B's in_seq), so that starting from that point, if A tries to do a reconnect, it will crash.

The next question is, why B's in_seq is a very large number even after rebinding?

#6 - 11/27/2014 12:02 AM - Guang Yang

The next question is, why B's in_seq is a very large number even after rebinding?

After more deep dive, I think the fix is that, if there is a hard reset from one side (rebind, clearing everything), the other side should reset in_seq to 0 to keep consistency, otherwise: 1) some messages might be dropped wrongly, 2) OSD crash if there is a reconnection attempt.

Thoughts?

#7 - 11/27/2014 01:16 AM - Haomai Wang

AFAR, if server rebind and client try to connect, client won't get CEPH_MSGR_TAG_SEQ tag from server because no replacing produce happened so it won't result in assert failed.

#8 - 11/27/2014 03:41 AM - Guang Yang

Update more logs from the **crashed** OSD:

```
2014-10-28 20:25:25.645004 7f288362e700 5 osd.483 15401 from dead osd.509, marking down, msg was 10.214.143.136:6831/5028784 expected :/0
2014-10-28 20:25:25.645122 7f288362e700 3 osd.483 15401 handle_osd_map epochs [15398,15398], i have 15401, src has [1,15398]
2014-10-28 20:25:25.648843 7f288362e700 5 osd.483 15401 from dead osd.509, marking down, msg was 10.214.143.136:6831/5028784 expected :/0
2014-10-28 20:25:25.648939 7f288362e700 3 osd.483 15401 handle_osd_map epochs [15398,15398], i have 15401, src has [1,15398]
2014-10-28 20:25:25.651019 7f288362e700 5 osd.483 15401 from dead osd.509, marking down, msg was 10.214.143.136:6831/5028784 expected :/0
2014-10-28 20:25:25.651134 7f288362e700 3 osd.483 15401 handle_osd_map epochs [15398,15398], i have 15401, src has [1,15398]
2014-10-28 20:25:25.652856 7f288362e700 5 osd.483 15401 from dead osd.509, marking down, msg was 10.214.143.136:6831/5028784 expected :/0
```

It seems that the peer OSD was marked down so that the Pipe with it was marked down (cleared). In this case, when peer issue a re-connect, how about doing a session reset (CEPH_MSGR_TAG_RESETSESSION) since we have cleared everything?

#9 - 11/27/2014 05:12 PM - Guang Yang

Guang Yang wrote:

Update more logs from the **crashed** OSD:
[...]

It seems that the peer OSD was marked down so that the Pipe with it was marked down (cleared). In this case, when peer issue a re-connect, how about doing a session reset (CEPH_MSGR_TAG_RESETSSESSION) since we have cleared everything?

CEPH_MSGR_TAG_RESETSSESSION properly is not the right way since it clears everything, it seems we should tell the peer to reset in_seq in this case?

#10 - 11/28/2014 12:57 AM - Guang Yang

Pull request - <https://github.com/ceph/ceph/pull/3029>

#11 - 12/01/2014 05:37 PM - Greg Farnum

If a connection gets marked down, we **cannot** reconnect to that endpoint again; it needs to recycle itself to a new entity. We've had several bugs in which the marked_down endpoint tries to reconnect and something isn't handled properly and this might be another one.

I want to do <http://fpaste.org/155680/48370014/>, but I've wanted to revert parts of this commit several times and been wrong...let's wait and hear from Sage before trying that out.

#12 - 12/01/2014 05:47 PM - Greg Farnum

- Assignee set to Greg Farnum

I'll turn that fpaste into a real patch and get Sam or somebody to put it in some testing so we should at least see if it unexpectedly breaks things before we merge and then backport it.

#13 - 12/02/2014 09:32 PM - Greg Farnum

- Status changed from New to 7

<https://github.com/ceph/ceph/pull/3070>

#14 - 12/11/2014 01:51 PM - Greg Farnum

- Status changed from 7 to Pending Backport

- Priority changed from High to Normal

Merged to master in commit:81af47531822343bc39ced6e4d9bad72ac982439

I guess it should get backported at some point, too, but we'll let it run in testing for a while first.

#15 - 02/11/2015 10:44 PM - Greg Farnum

- Backport set to *giant, firefly, dumpling*

#16 - 03/09/2015 08:10 PM - Loïc Dachary

- firefly backport <https://github.com/ceph/ceph/pull/3915>

#17 - 03/19/2015 09:05 AM - Loïc Dachary

- giant backport <https://github.com/ceph/ceph/pull/4089>

#18 - 03/26/2015 03:26 PM - Loïc Dachary

- Backport changed from *giant, firefly, dumpling* to *giant, firefly*

#19 - 03/26/2015 03:27 PM - Loïc Dachary

commit:1ccf583 SimpleMessenger: allow RESETSESSION whenever we forget an endpoint (in *giant*), commit:7ed92f7 SimpleMessenger: allow RESETSESSION whenever we forget an endpoint (in *firefly*),

#20 - 03/26/2015 03:27 PM - Loïc Dachary

- Status changed from *Pending Backport* to *Resolved*

#21 - 03/12/2019 11:12 PM - Greg Farnum

- Project changed from *Ceph* to *Messengers*

- Category deleted (*msg*)