

## fs - Bug #43960

### MDS: incorrectly issues Fc for new opens when there is an existing writer

02/04/2020 03:33 PM - Greg Farnum

<b>Status:</b>	New	<b>% Done:</b>	0%
<b>Priority:</b>	Urgent		
<b>Assignee:</b>	Greg Farnum		
<b>Category:</b>	Performance/Resource Usage		
<b>Target version:</b>	v16.0.0		
<b>Source:</b>	Development	<b>Affected Versions:</b>	
<b>Tags:</b>		<b>ceph-qa-suite:</b>	
<b>Backport:</b>	octopus,nautilus	<b>Component(FS):</b>	MDS
<b>Regression:</b>	No	<b>Labels (FS):</b>	
<b>Severity:</b>	3 - minor	<b>Pull request ID:</b>	
<b>Reviewed:</b>		<b>Crash signature:</b>	

#### Description

Cloned from [#43748](#), to cover the MDS-side issue. (Note that I have changed much of the text below to correct a few details.) In an active/standby configuration of two clients managed by file locks, the standby client causes unbuffered I/O on the active client, which is to be expected if they both ask for Frw. Additionally, and more importantly, failure of the standby client causes the active client I/O to freeze because the standby client somehow receives Fc caps it cannot release.

This can be reproduced by starting two ceph-fuse clients. Then in the first "active" client, do:

```
flock -x bar1 -c 'date; echo enter; while sleep 1; do date; echo foobar | tee -a bar1; done'
```

Then in the "standby" client:

```
flock -x bar1 -c 'date; echo 2 enter; sleep 30; date; echo 2 bye;'
```

Note: the second client should never print output because it cannot obtain the exclusive lock on bar1.

Now, send SIGKILL to the standby ceph-fuse client. This will cause I/O to halt for the first client until the MDS autocloses the "standby" ceph-fuse client session after 60s.

Here's the log from the standby client:

```
2020-01-21T17:44:33.248+0000 7f1d0b7fe700 20 client.4278 trim_cache size 2 max 16384
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 <== mds.0 v2:127.0.0.1:6854/
1480111679 45 ==== client_caps(revoke ino 0x10000000001 11 seq 34 caps=pAsLsXs dirty=- wanted=pFsc
r follows 0 size 294/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.297260+0000) v11 ==== 2
52+0+0 (crc 0 0 0) 0x7f1d1c00a1e0 con 0x557c4d018dc0
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 mds.0 seq now 31
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 5 client.4278 handle_cap_grant on in 0x10000000001 mds.
0 seq 34 caps now pAsLsXs was pAsLsXsFr
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 update_inode_file_time 0x10000000001.head
(faked_ino=0 ref=4 ll_ref=2 cap_refs={} open={1=1} mode=100640 size=287/0 nlink=1 btime=0.000000 m
time=2020-01-21T17:44:31.232141+0000 ctime=2020-01-21T17:44:31.232141+0000 caps=pAsLsXsFr(0=pAsLsX
sFr) objectset[0x10000000001 ts 0/0 objects 0 dirty_or_tx 0] parents=0x10000000000.head["bar1"] 0x
7f1cf4005a10) pAsLsXsFr ctime 2020-01-21T17:44:32.297260+0000 mtime 2020-01-21T17:44:32.297260+000
0
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 revocation of Fr
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 check_caps on 0x10000000001.head(faked_in
```

o=0 ref=4 ll\_ref=2 cap\_refs={} open={1=1} mode=100640 size=287/0 nlink=1 btime=0.000000 mtime=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXs(0=pAsLsXs) objectset[0x10000000001 ts 0/0 objects 0 dirty\_or\_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005a10) wanted pFscr used - issued pAsLsXs revoking Fr flags=0  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 cap mds.0 issued pAsLsXs implemented pAsLsXsFr revoking Fr  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 completed revocation of Fr  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 send\_cap 0x10000000001.head(faked\_ino=0 ref=4 ll\_ref=2 cap\_refs={} open={1=1} mode=100640 size=287/0 nlink=1 btime=0.000000 mtime=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXs(0=pAsLsXs) objectset[0x10000000001 ts 0/0 objects 0 dirty\_or\_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005a10) mds.0 seq 34 used - want pFscr flush - retain pAsLsXsXsFsxswbl held pAsLsXsFr revoking Fr dropping -  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 15 client.4278 auth cap, setting max\_size = 0  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 --> [v2:127.0.0.1:6854/1480111679,v1:127.0.0.1:6855/1480111679] -- client\_caps(update ino 0x10000000001 11 seq 34 caps=pAsLsXs dirty=- wanted=pFscr follows 0 size 287/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.297260+0000) v11 -- 0x7f1d1c009850 con 0x557c4d018dc0  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 <== mds.0 v2:127.0.0.1:6854/1480111679 46 ==== client\_caps(grant ino 0x10000000001 11 seq 35 caps=pAsLsXsFc dirty=- wanted=pFscr follows 0 size 294/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.297260+0000) v11 ==== 252+0+0 (crc 0 0 0) 0x7f1d1c00a1e0 con 0x557c4d018dc0  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 mds.0 seq now 32  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 5 client.4278 handle\_cap\_grant on in 0x10000000001 mds.0 seq 35 caps now pAsLsXsFc was pAsLsXs  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 update\_inode\_file\_time 0x10000000001.head(faked\_ino=0 ref=4 ll\_ref=2 cap\_refs={} open={1=1} mode=100640 size=287/0 nlink=1 btime=0.000000 mtime=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXs(0=pAsLsXs) objectset[0x10000000001 ts 0/0 objects 0 dirty\_or\_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005a10) pAsLsXs ctime 2020-01-21T17:44:32.297260+0000 mtime 2020-01-21T17:44:32.297260+0000  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 size 287 -> 294  
2020-01-21T17:44:33.343+0000 7f1d0a7fc700 10 client.4278 grant, new caps are Fc  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 <== mds.0 v2:127.0.0.1:6854/1480111679 47 ==== client\_caps(revoke ino 0x10000000001 11 seq 36 caps=pAsLsXs dirty=- wanted=pFscr follows 0 size 294/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.297260+0000) v11 ==== 252+0+0 (crc 0 0 0) 0x7f1d1c009850 con 0x557c4d018dc0  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 10 client.4278 mds.0 seq now 33  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 5 client.4278 handle\_cap\_grant on in 0x10000000001 mds.0 seq 36 caps now pAsLsXs was pAsLsXsFc  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 10 client.4278 update\_inode\_file\_time 0x10000000001.head(faked\_ino=0 ref=4 ll\_ref=2 cap\_refs={} open={1=1} mode=100640 size=294/0 nlink=1 btime=0.000000 mtime=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXsFc(0=pAsLsXsFc) objectset[0x10000000001 ts 0/0 objects 0 dirty\_or\_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005a10) pAsLsXsFc ctime 2020-01-21T17:44:32.297260+0000 mtime 2020-01-21T17:44:32.297260+0000  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 10 client.4278 revocation of Fc  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 10 client.4278 check\_caps on 0x10000000001.head(faked\_ino=0 ref=4 ll\_ref=2 cap\_refs={} open={1=1} mode=100640 size=294/0 nlink=1 btime=0.000000 mtime=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXs(0=pAsLsXs) objectset[0x10000000001 ts 0/0 objects 0 dirty\_or\_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005a10) wanted pFscr used - issued pAsLsXs revoking Fc flags=0  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 10 client.4278 cap mds.0 issued pAsLsXs implemented pAsLsXsFc revoking Fc  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 10 client.4278 completed revocation of Fc  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 10 client.4278 send\_cap 0x10000000001.head(faked\_ino=0 ref=4 ll\_ref=2 cap\_refs={} open={1=1} mode=100640 size=294/0 nlink=1 btime=0.000000 mtime=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXs(0=pAsLsXs) objectset[0x10000000001 ts 0/0 objects 0 dirty\_or\_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005a10) mds.0 seq 36 used - want pFscr flush - retain pAsLsXsXsFsxrwl held pAsLsXsFc revoking Fc dropping -  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 15 client.4278 auth cap, setting max\_size = 0  
2020-01-21T17:44:33.347+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 --> [v2:127.0.0.1:6854/1480111679,v1:127.0.0.1:6855/1480111679] -- client\_caps(update ino 0x10000000001 11 seq 36 caps=pAsLsXs dirty=- wanted=pFscr follows 0 size 294/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.297260+0000) v11 -- 0x7f1d1c009dd0 con 0x557c4d018dc0  
2020-01-21T17:44:33.349+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 <== mds.0 v2:127.0.0.1:6854/1480111679 48 ==== client\_caps(grant ino 0x10000000001 11 seq 37 caps=pAsLsXsFrw dirty=- wanted=pFscr follows 0 size 294/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.297260+0000) v11 ==== 252+0+0 (crc 0 0 0) 0x7f1d1c009dd0 con 0x557c4d018dc0

```

2020-01-21T17:44:33.349+0000 7f1d0a7fc700 10 client.4278 mds.0 seq now 34
2020-01-21T17:44:33.349+0000 7f1d0a7fc700 5 client.4278 handle_cap_grant on in 0x10000000001 mds.
0 seq 37 caps now pAsLsXsFrw was pAsLsXs
2020-01-21T17:44:33.349+0000 7f1d0a7fc700 10 client.4278 update_inode_file_time 0x10000000001.head
(faked_ino=0 ref=4 ll_ref=2 cap_refs={} open={1=1} mode=100640 size=294/0 nlink=1 btime=0.000000 m
time=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXs(0=pAsLsXs)
objectset[0x10000000001 ts 0/0 objects 0 dirty_or_tx 0] parents=0x10000000000.head["bar1"] 0x7f1c
f4005a10) pAsLsXs ctime 2020-01-21T17:44:32.297260+0000 mtime 2020-01-21T17:44:32.297260+0000
2020-01-21T17:44:33.349+0000 7f1d0a7fc700 10 client.4278 grant, new caps are Frw
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 <== mds.0 v2:127.0.0.1:6854/
1480111679 49 ==== client_caps(revoke ino 0x10000000001 11 seq 38 caps=pAsLsXsFr dirty=- wanted=pF
scr follows 0 size 294/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.297260+0000) v11 ====
252+0+0 (crc 0 0 0) 0x7f1d1c004260 con 0x557c4d018dc0
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 10 client.4278 mds.0 seq now 35
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 5 client.4278 handle_cap_grant on in 0x10000000001 mds.
0 seq 38 caps now pAsLsXsFrw was pAsLsXsFrw
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 10 client.4278 update_inode_file_time 0x10000000001.head
(faked_ino=0 ref=4 ll_ref=2 cap_refs={} open={1=1} mode=100640 size=294/0 nlink=1 btime=0.000000 m
time=2020-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXsFrw(0=pAsLs
XsFrw) objectset[0x10000000001 ts 0/0 objects 0 dirty_or_tx 0] parents=0x10000000000.head["bar1"]
0x7f1cf4005a10) pAsLsXsFrw ctime 2020-01-21T17:44:32.297260+0000 mtime 2020-01-21T17:44:32.297260+
0000
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 10 client.4278 revocation of Fw
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 10 client.4278 check_caps on 0x10000000001.head(faked_in
o=0 ref=4 ll_ref=2 cap_refs={} open={1=1} mode=100640 size=294/0 nlink=1 btime=0.000000 mtime=2020
-01-21T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXsFr(0=pAsLsXsFr) obje
ctset[0x10000000001 ts 0/0 objects 0 dirty_or_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005
a10) wanted pFscr used - issued pAsLsXsFr revoking Fw flags=0
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 10 client.4278 cap mds.0 issued pAsLsXsFr implemented p
AsLsXsFrw revoking Fw
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 10 client.4278 completed revocation of Fw
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 10 client.4278 send_cap 0x10000000001.head(faked_ino=0 r
ef=4 ll_ref=2 cap_refs={} open={1=1} mode=100640 size=294/0 nlink=1 btime=0.000000 mtime=2020-01-2
1T17:44:32.297260+0000 ctime=2020-01-21T17:44:32.297260+0000 caps=pAsLsXsFr(0=pAsLsXsFr) objectset
[0x10000000001 ts 0/0 objects 0 dirty_or_tx 0] parents=0x10000000000.head["bar1"] 0x7f1cf4005a10)
mds.0 seq 38 used - want pFscr flush - retain pAsLsXsFrw held pAsLsXsFrw revoking Fw droppi
ng -
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 15 client.4278 auth cap, setting max_size = 0
2020-01-21T17:44:33.350+0000 7f1d0a7fc700 1 -- 127.0.0.1:0/361750000 --> [v2:127.0.0.1:6854/14801
11679,v1:127.0.0.1:6855/1480111679] -- client_caps(update ino 0x10000000001 11 seq 38 caps=pAsLsXs
Fr dirty=- wanted=pFscr follows 0 size 294/0 ts 1/18446744073709551615 mtime 2020-01-21T17:44:32.2
97260+0000) v11 -- 0x7f1d1c009dd0 con 0x557c4d018dc0

```

There's a lot of back-and-forth on the caps but the highlight here to me is that "grant, new caps are Fc". The standby client first notes it wants those caps as part of opening the file just prior to setfilelock, on the MDS:

```

2020-01-21T17:44:25.829+0000 7f651abf0700 7 mds.0.server dispatch_client_request client_request(c
lient.4278:7 open #0x10000000001 2020-01-21T17:44:25.829383+0000 caller_uid=1156, caller_gid=1156{
1001,1156,}) v4
...
2020-01-21T17:44:25.830+0000 7f651abf0700 20 mds.0.locker client.4278 pending pAsLsXs allowed pAs
LsXs wanted pFscr
2020-01-21T17:44:25.830+0000 7f651abf0700 10 mds.0.locker eval done
2020-01-21T17:44:25.830+0000 7f651abf0700 12 mds.0.server open issued caps pAsLsXs for client.4278
on [inode 0x10000000001 [2,head] /foo/bar1 auth v96 ap=1 dirtyparent s=252 n(v0 rc2020-01-21T17:4
4:25.531606+0000 b252 1=1+0) (ifile excl->sync) (iversion lock) cr={4274=0-4194304@1} caps={4274=p
AsLsXsFscr/pAsLsXsFsxcrrw/pFscr@9,4278=pAsLsXs/pFscr@3},l=4274(-1) | ptrwaiter=0 request=1 lock=1
caps=1 dirtyparent=1 dirty=1 waiter=0 authpin=1 0x55f2eda8c800]
2020-01-21T17:44:25.830+0000 7f651abf0700 20 mds.0.bal hit_dir 0 pop is 1.99945, frag * size 1 [po
p IRD:[C 2.00e+00] IWR:[C 9.31e+00] RDR:[C 0.00e+00] FET:[C 0.00e+00] STR:[C 0.00e+00] *LOAD:20.6]
2020-01-21T17:44:25.830+0000 7f651abf0700 7 mds.0.server reply_client_request 0 ((0) Success) cli
ent_request(client.4278:7 open #0x10000000001 2020-01-21T17:44:25.829383+0000 caller_uid=1156, cal
ler_gid=1156{1001,1156,}) v4
...

```

```
2020-01-21T17:44:25.833+0000 7f651abf0700 1 -- [v2:127.0.0.1:6854/1480111679,v1:127.0.0.1:6855/1480111679] <== client.4278 127.0.0.1:0/361750000 16 ==== client_request(client.4278:8 setfilelock rule 2, type 2, owner 17160986462880593306, pid 0, start 0, length 0, wait 1 #0x10000000001 2020-01-21T17:44:25.832830+0000 caller_uid=1156, caller_gid=1156{1001,1156,}) v4 ==== 152+0+0 (crc 0 0 0) 0x55f2edb8ec00 con 0x55f2edb31680
```

So, why on earth is the MDS issuing Fc to the standby client when the already-active client wants Fw? Perhaps it's evaluating based on what is actually granted?

**Related issues:**

Copied from fs - Bug #43748: client: improve wanted handling so we don't requ...

**Fix Under Review**

**History**

**#1 - 02/04/2020 03:33 PM - Greg Farnum**

- Copied from Bug #43748: client: improve wanted handling so we don't request unused caps (active-standby exclusive file lock case) added

**#2 - 02/04/2020 04:04 PM - Greg Farnum**

From the original ticket:

Jeff Layton wrote:

Greg pointed out some things in a face-to-face discussion the other day that lead me to question whether this ought to be a client-side bug.

[...]

Why are Frw being revoked here? We aren't changing the layout of the file, so that seems a bit strange.

There may be some routine things happening that will do this periodically and briefly (I think some metadata flushes such as mtime technically require this, although we've relaxed consistency for most of them IIRC), but I agree that in this context it looks strange.

It's probably fine to hand out Fc caps to a singleton client that has the file open O\_RDWR, but once the second client opens the file with O\_RDWR (or O\_WRONLY) Fc should be revoked and should not be handed out again until there aren't multiple writers.

ISTM that the client ought to be free to set "wanted" in its mask as it sees fit without much regard for other clients, and that it's up to the MDS to mediate "disputes" between them.

Yes, that all sounds correct to me!

#3 - 02/06/2020 12:11 AM - Greg Farnum

Okay, I dove into this a bit today. No final conclusions but reminding myself about how some of this works and several points of interest: As written, that flock command involves opening the bar1 file, checking the size, and appending to the end of it every second. On the MDS, part of that process is a lookup op which is responsible for most of the cap churn — the MDS has to revoke Fw from the "primary" client, then issues it back Fscr as part of the lookup reply. But then for some reason (not fully explored) the inode filelock gets moved into the LOCK\_SYNC\_LOCK state, and that only allows Fc caps — so the primary client gets a revoke of Fsr, but the secondary client gets a **grant** of Fc even though it didn't start with them (they are in the wanted set)! I'm not certain if this harms anything directly (at the end of this process the secondary end up with Fr) but it certainly doesn't help.

When you kill the secondary, it holds some cap and isn't available to acknowledge the revoke, so you have to wait through the blacklist timeout. There's not a great solution to this other than somehow blocking it from holding any caps at all...

=====

However, things are quite different if we instead set the tee operation to merely overwrite. We still get the 1-minute hang when killing the secondary process, but the cap cycling is a lot simpler and might be easier to adjust/fix. I believe the differences are rooted in the active client not needing to get a file size, but merely issuing blind writes to offset 0. It looks like in that case, there is a small cap loop on the events but the main problem is that the secondary client wants Fscr, yet is GRANTED Frw as part of the active client's updates! I haven't diagnosed exactly what's going on there but it looks like the CInode::get\_caps\_wanted() interfaces are incorrectly compressing the client wanted caps into a single mask which is then used to grant caps back out to everybody with **any** wanted bits.

#4 - 02/06/2020 03:06 PM - Zheng Yan

Greg Farnum wrote:

Okay, I dove into this a bit today. No final conclusions but reminding myself about how some of this works and several points of interest: As written, that flock command involves opening the bar1 file, checking the size, and appending to the end of it every second. On the MDS, part of that process is a lookup op which is responsible for most of the cap churn — the MDS has to revoke Fw from the "primary" client, then issues it back Fscr as part of the lookup reply. But then for some reason (not fully explored) the inode filelock gets moved into the LOCK\_SYNC\_LOCK state, and that only allows Fc caps — so the primary client gets a revoke of Fsr, but the secondary client gets a **grant** of Fc even though it didn't start with them (they are in the wanted set)! I'm not certain if this harms anything directly (at the end of this process the secondary end up with Fr) but it certainly doesn't help.

When you kill the secondary, it holds some cap and isn't available to acknowledge the revoke, so you have to wait through the blacklist timeout. There's not a great solution to this other than somehow blocking it from holding any caps at all...

=====

However, things are quite different if we instead set the tee operation to merely overwrite. We still get the 1-minute hang when killing the secondary process, but the cap cycling is a lot simpler and might be easier to adjust/fix. I believe the differences are rooted in the active client not needing to get a file size, but merely issuing blind writes to offset 0. It looks like in that case, there is a small cap loop on the events but the main problem is that the secondary client wants Fscr, yet is GRANTED Frw as part of the active client's updates! I haven't diagnosed exactly what's going on there but it looks like the CInode::get\_caps\_wanted() interfaces are incorrectly compressing the client wanted caps into a single mask which is then used to grant caps back out to everybody with **any** wanted bits.

It's because of CInode::get\_caps\_liked(). mds always issue any file caps allowed lock state to client.

**#5 - 02/10/2020 02:53 PM - Patrick Donnelly**

- Assignee set to *Greg Farnum*

**#6 - 02/26/2020 05:38 PM - Patrick Donnelly**

- Target version changed from *v15.0.0* to *v16.0.0*

- Backport set to *octopus,nautilus*