

Courtesy Locks in NFSv4.0

Before BAT

Fall Bakeathon, 2010

October 6, 2010

Dave Noveck

Background

- Courtesy locks concept in RFC3530
 - Allows reduction of client disruption when there are network problems (esp. intermittent)
 - Optional for server
 - Not RECOMMENDED, but who's going to be against "courtesy"
- Not very clearly defined in RFC3530
 - Need to clarify
 - RFC5661 has more detail
 - But also uses protocol elements v4.0 doesn't have

Major Questions

- When there is a conflict, what is the scope of lock set gotten rid of
- For locks gotten rid of, how is the loss communicated to client.
- If via NFS4ERR_EXPIRED, how does that relate to client handling of NFS4ERR_EXPIRED

Conflict scope

- All locks for client
 - Gives 90% of the benefit
 - Avoids other issues
 - But not clear that is what spec intended
- All locks for client/fh pair
 - Avoid lock pairing rules
- Something smaller
 - Need to look at lock failure dependency as is done by RFC5661

What error to return?

- NFS4ERR_EXPIRED seems sensible
 - But some client may assume all locks lost and do reboot equivalent
 - Should we say not do that
 - Or such clients compliant but ill-equipped to deal with courtesy
- NFS4ERR_ADMIN_REVOKED
 - Not exactly true but practical
- Other choices?

How to recover?

- Don't have some things from v4.1
 - TEST_STATEID. FREE_STATEID
- Absence of TEST_STATEID
 - Stateids can be test but takes longer
 - One stateid per OP
 - COMPOUNDS can be used but stop when a lost lock is found.
- Absence of FREE_STATEID
 - Stateids are invalid and get error if referenced
 - Exception for CLOSE?
 - More coarse-grained lock failure but who cares very much
 - Could rely on server to free them after a long while

Going forward

- Need to find out what people currently do
 - Thanks for responses to questionnaire received so far
 - Others, consider yourself nagged
- Once we get a more complete response set, I'll make a proposal
 - Appreciate discussion and suggestions both before the proposal and after
 - Goal is to clear up uncertainty and get something that can be incorporated in RFC3530bis

Courtesy Locks in NFSv4.0

After BAT

Fall Bakeathon, 2010

Corrected and updated to reflect discussion on

October 6, 2010

Dave Noveck

Background

- Courtesy locks concept in RFC3530
 - Allows reduction of client disruption when there are network problems (esp. intermittent)
 - Optional for server
 - Not RECOMMENDED, but who's going to be against "courtesy"
- Not very clearly defined in RFC3530
 - Need to clarify
 - RFC5661 has more detail
 - But also uses protocol elements v4.0 doesn't have
- Corrections and additions in green

Major Questions

- When there is a conflict, what is the scope of lock set gotten rid of
- For locks gotten rid of, how is the loss communicated to client.
- If via NFS4ERR_EXPIRED, how does that relate to client handling of NFS4ERR_EXPIRED

Conflict scope

- All locks for client
 - Gives 90% of the benefit
 - Avoids other issues
 - But not clear that is what spec intended
- All locks for client/fh pair
 - Avoid lock pairing rules
- Something smaller
 - Need to look at lock failure dependency as is done by RFC5661

What error to return?

- NFS4ERR_EXPIRED seems sensible
 - But some client may assume all locks lost and do reboot equivalent
 - Should we say not **to** do that
 - Or **we could define** such clients **as** compliant but ill-equipped to deal with courtesy
- NFS4ERR_ADMIN_REVOKED
 - Not exactly true but practical
- Other choices?

How to recover?

- Don't have some things from v4.1
 - TEST_STATEID. FREE_STATEID
- Absence of TEST_STATEID
 - Stateids can be test but takes longer
 - One stateid per OP
 - COMPOUNDS can be used but stop when a lost lock is found.
- Absence of FREE_STATEID
 - Stateids are invalid and get error if referenced
 - Exception for CLOSE?
 - More coarse-grained lock failure but who cares very much
 - Could rely on server to free them after a long while

Going forward

- Need to find out what people currently do
 - Thanks for responses to questionnaire received so far
 - Others, consider yourself nagged
 - Consider yourself nagged again
 - Need responses to go forward with this
- Once we get a more complete response set, I'll make a proposal
 - Appreciate discussion and suggestions both before the proposal and after
 - Goal is to clear up uncertainty and get something that can be incorporated in RFC3530bis