LAYOUTCOMMIT for pNFS file

Fall 2010 Bakeathon
Corrected and updated to reflect discussion on
October 4, 2010
Dave Noveck

Outline (Talk + Sequel)

- Fundamentals of LAYOUTCOMMIT
  - Why is it there?
  - What problems does it solve?
- LAYOUTCOMMIT (for file) details
  - One approach (i.e. mine)
  - Other issues that arose in discussion in green
- Corrections in green
- Going forward
  - Getting group agreement (on something)
  - Clarifying/updating documents

Basic pNFS Issue

- Fundamental pNFS premise
  - pNFS splits data and metadata
  - But, changing data requires metadata change
    - E.g. size, modified time, change
- Alternative responses
  - Give up on pNFS premise
    - Only truly principled response :-)
    - But there is this thing called performance
    - And the issue is **small,** we assure ourselves. (And it really is)
  - Look in designers' back of tricks (See later slides)

Things to keep in mind

- Solution must support all layout types
  - Including those not invented yet
  - Lots of people say “#$%#, gimme a beak”
- Update semantics need to be considered
  - Perfect instantaneous coherence is *terrific*
    - May be unbearably complicated/expensive
    - Most applications don’t need that
    - Writer knows he wrote it
    - Others are not synchronized with writer (so who cares?)

Some Important Disagreements

- Big issue about cost of LAYOUTCOMMIT

1. Don’t worry it’s trivial (or can be made so)
2. May be significant so protocol should define minimum needed

- Relation between COMMIT and LAYOUTCOMMIT

1. Must do after each COMMIT
2. Or set of COMMITS
3. Must do LAYOUTCOMMIT before flushing written pages from cache.
4. Disagreement: what is the reason for all that?
5. Note that Sync. WRITE equal Async. WRITE plus COMMIT

Data server responsibility?

- Require DS to update MDS appropriately
  - What does “appropriately” mean
  - WG could spend a while figuring out
  - “rough consensus” might never happen
  - Won’t work for pNFS block
- Client is to update MDS appropriately
  - Thus is born LAYOUTCOMMIT
  - Still have issue of “appropriately”

LOC for file: Structural Issues

- For pNFS block, LAYOUTCOMMIT cannot be avoided
- For pNFS file, it is more like an optimization
- Tend to think of LAYOUTCOMMIT as license for lack of attribute coherence
  - Not wrong but not only way to think about it

LOC for file: Practical issues

- When must client do it?
- What if client doesn’t?
  - Relates primarily to client/network failure
- Role of CLOSE-to-OPEN consistency
  - Part of the protocol?
  - Spec says CLOSE-to-OPEN is supported
  - Can clients get more consistency?
  - Are they allowed to get by on less?

LOC for file: Start with Proposal

- Need to start discussion somewhere
  - Will offer my approach
  - As a way to start discussion
  - Even though I know my approach must be right :-)
- Basic approach:
  - Use optimization paradigm
  - C-to-O consistency is a common choice
    - Not a requirement
    - But is default behavior (client and network failure)

LOC for file: My Answers (1)

- When must client do it?
  - Whenever client wants
  - On CLOSE?
    - If client want to?
    - Or treat a set of OPEN/CLOSEs as a unit
    - Up to client
- Server MAY do attribute updates on CLOSE
  - But not a requirement

LOC for file: My Answers (2)

- What if client doesn’t?
  - Server MAY update attributes based on IO
  - Coherent distributed FS will work
  - No requirement to not update based on absence of LOC
- Role of CLOSE-to-OPEN consistency
  - Not part of the protocol
  - Clients can get more consistency or less.

LOC for file: My Answers (3)

- Servers (as a unit) MUST provide for client unable to do LOC (or can’t see it if we did)
  - Don’t know if he would have
  - Assume he would have, LOC equivalent
    - OPEN lost due to lease expiration
    - Client reboot
- What about DS-MDS disconnect?
  - MUST assume worst-case: if any write done to DS, LOC-equivalent done
  - MDS must know about LOC-pending state before it becomes effective

Additional Issue

- Periodic LAYOUTCOMMIT’s required
- Suppose a file open for days/weeks.
- Periodic WRITEs
- If no LAYOUTCOMITs, attributes out of date
- Is there a need/requirrment for period LAYOUTCOMMITs in this case
  - What would the frequency be?
  - Lease time?

After

- Questions, criticism, discussion
- Subsequent discussion on list
- Try to reach *Consistent sense of the group*
- Document possibilities
  - Errata
  - Short internet draft clarifying/correcting RFC
  - Decide spec is OK
    - And it only needs an I-D with implementation advice