Possible Addition of Directory Layouts

Exploring One Way of Improving Parallelism of Directory Operations

David Noveck

April 29, 2020

Post-IETF107 Virtual WG Meeting

Overview

- Basic idea: pNFS-like Directory Layouts *without striping*
- Why this might be worth following up on.
- Previous related work
- Alternatives
- Open issues to resolve
- Going forward (or not)

Basic Idea

pNFS-like Directory Layout (without striping)

- Provides greater parallelism in the handling of directory operations.
  - Inter-directory parallelism without intra-directory parallelism.
- pNFS-*like* (Commonalities and Differences)
  - Primary responsibility for a function given to another server
  - Layout bestower can perform the function if requested
  - No Striping
  - But you would have at least read and write layouts
  - Only a single layout type likely

Why this is Worth Exploring

Relatively Easy Way of Improving Directory Parallelism

- Need to provide greater parallelism in handling of directory operations
  - Focusing on build environments, rather than troublesome cases such as untar.
  - Rm –r is of some, relatively, minor interest
  - Assuming enough jobs to avoid need to parallelize single cmds.
- Many attempts have not resulted in improvements:
  - Attempts to support directory striping have not worked out. See Next Slide.
  - Directory delegations and notifications in RFC5661 but not implemented. Not clear why.

Previous Related Work

pNFS-like Directory Layout (with striping)

- More natural fit for pNFS model, but …
  - Striping is difficult for directories
    - No obvious correlate for file offset
    - Various hashes might be used.
    - But it is hard to get agreement among client, server, on-disk fs.
  - Necessary for performance of extremely large directories
    - Not clear how common these are
- Good way to provide within-directory parallelism
  - Better than only providing inter-directory parallelism, but history so far is not encouraging.

Possible Alternatives

Directory Delegations and Other Similar Ideas

- Directory Delegations
  - Not precisely an alternative, but …
    - Addresses performance of many of the same workloads.
    - Inevitably this will result in conflicts for resources.
  - In RFC5661 but never implemented. Need to understand why.
    - Might be simple inertia
    - Lack of performance improvement with file delegations might have a role
    - Need to understand if expectations for directory delegation are better
    - Is there a problem with directory delegations that we can address?
- Other possible alternatives:
  - If you know of any, need to discuss on list.

Issues to Resolve

Protocol Details to Work Out

- Protocol Details are now TBD
  - Easiest approach would be to reserve a mapping type for this use.
  - Would allow directory layouts to have their own definitions
  - Directory layouts with striping would get its mapping type
- Layout Levels
  - Need Read and Write
  - May need read/write bits for opens, based on what is denied

Issues to Resolve

Potentially Troublesome Interactions to Look At

- Issues with RENAME across directories
  - Should be OK if target if it has layout for both directories
  - If not, go to main metadata server.
  - When renaming a directory do not need a layout for it.
- Interaction with file delegations
  - Server with laout should be able to grant and recall deegations.
  - If layout recalled, main metadata server will become responsible,
- Interaction with directory delegations and notifications.
  - Even though there is a a potential resource conflict, spec will have to address interaction.

Going Forward (or Not)

Assessing Interest and Possible Next Steps

- Looking to Assess WG interest (Here and on list)
- If not much, want to understand why.
  - Just not a compelling approach?
  - Prefer other ways of dealing with issues?
  - Is directory operation performance not important enough?
- If there is significant interest, what are next steps?
  - I could produce a short I-D with enough detail to allow prototypes.
  - Other ideas?