Coherent data caching for NFS

IETF87

Marc Eshel, Manoj Naik, Dean Hildebrand

7/29/2013

Problem Statement

- Provide POSIX Read/Write Semantics
  - Better than NFS close-to-open
  - Allow applications to run unchanged over NFS
- Improve NFS client caching
  - Allow byte-range caching to reduce data revalidations
  - More coherent cache

Who needs this?

- Applications that require strong caching semantics
- HPC applications that work on segments of very big files
- Application that share files for synchronization and communication between process on different clients
- Shared append-only files, such as logs

Objective of this presentation

- Agree on the importance of the problem
- Start discussion on a high-level direction
  - Need POSIX semantics?
  - Need better caching?
  - Both?

POSIX Semantics

- Define what this is and how different from close-to-open
  - Requires that a read which can be proved to occur after a write has returned returns the new data.
- Applications currently need to use locking + direct IO, needs application change
- How to achieve - tokens, etc.
  - byte-range locking is too intrusive when sharing is rare

Input from mailing list

- There was a proposal from Trond and others to add byte range delegation
- We need Posix write semantics
- For Posix write use locks and direct IO
- Use tokens, we can call it something else but the concept is the same

Brent: if you are interested in more efficient cache consistency protocols,you want a callback scheme where the message overhead is proportional to the sharing that actually occurs. The byte-range locking protocol is instead proportional to the I/O that occurs.

Options

- Byte-range delegations (Trond+)
  - Lack of interest?
- When to use byte-range
  - NFSv4.2 hints
  - Mount options

Implementation needed

- Start with Trond's draft

  http://tools.ietf.org/id/draft-myklebust-nfsv4-byte-range-delegations-00.txt
- Call it tokens not delegations
- Update for NFSv4.1
  - Sessions
  - pNFS
- Request minimum and maximum range needed
- Align range to block boundary