



Go further, faster™

De-Duplication Awareness and Sub-file Caching

draft-eisler-nfsv4-pnfs-dedupe-01.txt

IETF-79

2010-11-09

Mike Eisler





draft-eisler-nfsv4-pnfs-dedupe-01.txt

- Differences from last update
- NFSv4.x or standalone?
- Key points solved
- Issues of contention
- Data that supports proposal
- Overlaps
- Should proposal be combined/not combined?
- Who should review?
- Next steps



draft-eisler-nfsv4-pnfs-dedupe-01.txt

- Differences from last update
 - Explicit support for de-dupe, sub-file caching with call backs, combined de-dupe, sub-file caching with call backs
 - Use of private layout types eliminated (unnecessary)
 - Explains how to use the proposed COPY operations to enable de-duplication on the WRITE path
- NFSv4.x or standalone?
 - READ-side of this is standalone: 192 distinct layout types
 - Requires NFSv4.1 or higher
- Key points solved
 - Allow NFS servers to communicate de-duplication maps to clients
 - conserves network and CPU bandwidth
 - Since a callback is used (CB_LAYOUTRECALL) sub-file caching is a side benefit



draft-eisler-nfsv4-pnfs-dedupe-01.txt

■ Issues of contention

- Support for blocks that are not powers of two
 - Turns out proposal will support that
 - However, server is not obliged to do so
 - I doubt any server would do so
- Differential compression: e.g. de-duping
 - block 1: “**abcd**” with
 - block 2: “**zabc**”

This is out of scope
- Combining with conventional (RAID0) pNFS types
 - Nothing prevents this
 - Client would obtain de-dupe/sub-file layout and a RAID0 layout type
 - Use former to decide what byte range of what file to get a layout for the latter

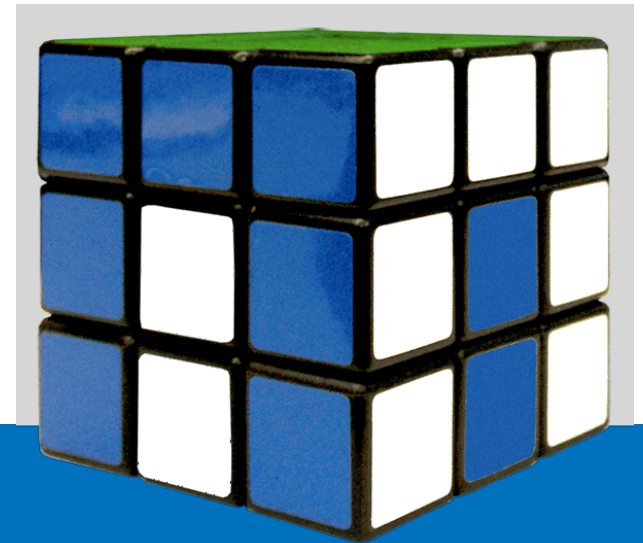


draft-eisler-nfsv4-pnfs-dedupe-01.txt

- Data that supports proposal
 - Clearly there is a need for this, or Dell would not have acquired Ocarina
 - Ocarina apparently exchanges block checksums
 - This approach considered and rejected because clients and servers would have to agree on a checksum algorithm
- Overlaps
 - draft-hildebrand-nfsv4-read-sparse-01.txt
 - Provides read de-dupe of holes
 - Effectively a sub-set of functionality of draft-eisler-nfsv4-pnfs-dedupe-01.txt
- Should proposal be combined/not combined?
 - Should not be combined with draft-hildebrand-nfsv4-read-sparse-01.txt
- Who should review?
 - WG
- Next steps
 - Complete recovery pieces; doable by IETF 80
 - Prototyping from NetApp unlikely to occur unless 3rd party ISVs show interest

Thanks

draft-eisler-nfsv4-pnfs-dedupe-01.txt



Questions and Answers