The pNFS NFS-Objects Layout

IETF-85
Atlanta, GA
2012-11-08

Benny Halevy <bhalevy@tonian.com>

Background

- The proposed NFS-Objects layout is based on the pNFS Objects layout (RFC5664)
- It reuses the Objects layout:
  - Flexible, per-file striping patterns,
  - Client-side object-based RAID,
  - Centralized security model
- And adds NFS as:
  - A storage-access protocol and
  - Basic back-end control protocol
  - Like the way T-10 OSD is currently being used

History

- RFC5664 defines the pNFS Objects Layout
- Ratified Jan. 2010
- RFC5664bis underway contains minor technical errata

Motivation

- **Use an ubiquitous, standard protocol to access the data servers**
  - The T10 OSD protocol, although standard, lacks wide adoption, while NFS is very popular and widely available
- **Encourage best-of-breed solutions**
  - Relying on NFS as the basic back-end control protocol allows one to mix and match a metadata server and data servers from different vendors

Use Cases

- **Aggregation of standalone NFS servers**
  - Customers heavily invested in NFSv3 filers, looking to reuse their legacy filers as data servers in a clustered configuration
- **Exporting of existing clustered file system**
  - For example: Ceph, Gluster
  - These file systems do not have a standard storage access protocol. Therefore NFS can be used instead

Use Cases (2)

- **Flexible, per-file striping patterns**
  - Application SLAs and management policies as well as dynamic load balancing and tiering decisions require per-file control over striping
  - Existing clustered FSs do not map to the files layout striping patterns

Security Model

- A simple security control method is proposed:
  - The metadata server controls the file ownership and permissions of the objects
  - Client is handed a corresponding RPC credential on LAYOUTGET a-la OSD capabilities
  - Outstanding credentials are unilaterally revoked by the MDS by modifying the objects user or group owner
- Data Servers should be configured to allow root access only to MDS

Security Considerations

- pNFS files layout provides tighter security and DS-based access enforcement (client fencing)
  - By using a proprietary back-end control protocol
- Block layout (RFC5663) has no such mechanism and respective caveats are documented there
- Objects layout (RFC5663) provides robust security and client fencing model using T-10 OSD
  - But lacks global state to enforce e.g. mandatory locking

Security Considerations (2)

- **Advisory security mechanism**: NFSv3 can be used to simulate the basic object capability model using a per-file rpc credentials handed out to clients as a naive RO or RW capability
- **Mandatory security mechanism**: NFSv4 and RPCSEC GSS can be used for authentication and authorization
- **Global State**: Support for back-end protocol a-la files layout is possible
  - OPEN at MDS, client is handed out a stateid to use.
  - For the mix-and-match design goal it is required to be a (IETF) standard