ORACLE

NFSv4 End-to-end Data Integrity

Chuck Lever
Consulting Member of Technical Staff

Data Corruption

Generic Causes

- Corruption can occur
  - During transmission
  - When storing data
  - While data is at rest
  - During retrieval
- Corruption can be
  - Active: data overwritten or misplaced
  - Passive: failure of physical device or firmware

Detected Or Silent?

"The goal of data integrity protection is not to make corruption impossible, but rather to ensure corruption is detected before it can no longer be corrected, or before corrupt data is used by an application."

End-to-End Data Integrity

Single-stage v. End-to-End

- Single-stage verification
  - Data may or may not be verified at each node in a data flow
  - Protection is isolated -- phone tag syndrome

- End-to-End verification
  - Integrity metadata is available at each step
  - A standard mechanism verifies data at each node
  - On write: data delivered to storage devices is guaranteed to be the same data provided by applications
  - On read: data delivered to applications is guaranteed to be the same data that resides in storage

End-to-End Data Integrity

Terminology

- Protection envelope
  - A set of nodes in an I/O system which together guarantee data integrity from input to output
- Protection information
  - Integrity metadata, possibly a checksum or something more complex
- Protection interval
  - Fixed-size portion of application data protected by one protection information field
- Protection type
  - An enumerated value indicating the size, contents, and interpretation of protection information fields

End-to-End Data Integrity

NFS Use Cases

- Purpose
  - Enable end-to-end data integrity between applications on NFS clients and block storage on NFS servers
- Potential applications
  - Virtual disk devices in hypervisor environments
  - Block-based database backends
- Where we are not going (yet)
  - Generic support for POSIX-compliant applications
  - Defining behavior of servers with regard to their integrity-enabled storage

End-to-End Data Integrity

Open Questions

- NFSv4 reboot recovery
- DS failover mechanisms
- Server-to-server and migration behavior
- Read-ahead
- Unstable writes

NFSv4 End-to-end Protocol

Protection type

- Enumerated integer values
  - enum nfs_protection_type4

- For example, the value “1” could represent T10 PI Type 1:
  - 512-byte protection interval
  - PI field is application-owned
  - 8-byte protection information field containing:
    - 2-byte guard tag (CRC-16 checksum of protection interval)
    - 2-byte application tag (user defined)
    - 4-byte reference tag (LO 32-bits of LBA)

NFSv4 End-to-end Protocol

Protection type discovery

- Client queries server to discover which protection types are supported
- Server can report that multiple protection types are supported
- XDR for new read-only per-FSID GETATTR attribute:
  - FATTR4_PROTECTION_TYPES = 82
  - nfs_protection_type4 fattr4_protection_types<>

NFSv4 End-to-end Protocol

Conveying data and protection information

- New arm of data_content4:
  - NFS4_CONTENT_PROTECTED_DATA
  - pd_type is the protection type in effect for this I/O
  - pd_info is a packed array of protection fields

```
struct data_protected4 {
    nfs_protection_type4 pd_type;
    offset4              pd_offset;
    bool                 pd_allocated;
    opaque               pd_info<>;
    opaque               pd_data<>;
};
```

NFSv4 End-to-end Protocol

Reporting integrity errors

- Additions to nfsstat4:
  - NFS4ERR_PROT_NOTSUPP - Server does not support specified protection type
  - NFS4ERR_PROT_INVAL - the protection field count does not match the amount of data in pd_data
  - NFS4ERR_PROT_FAIL - integrity verification failure on write
  - NFS4ERR_PROT_LATFAIL - integrity verification failure on read

NFSv4 End-to-end Protocol

Multi-server considerations

- The same set of protection types MUST be supported
  - By each DS in a layout
  - By each replica of the same file system
  - By the source and destination file servers during migration
- The same set of protection types MAY be supported
  - By both sides of a server-to-server copy
- Protection information MUST be copied
  - If both sides support the same protection type

Next Steps

- Iterate on end-to-end document
  - draft-cel-nfsv4-end2end-data-protection-00 (July 2)
- Identify reasonable test applications
- Prototype the proposed protocol
- Other proposals
  - Spencer's pNFS-based approach
  - Others?