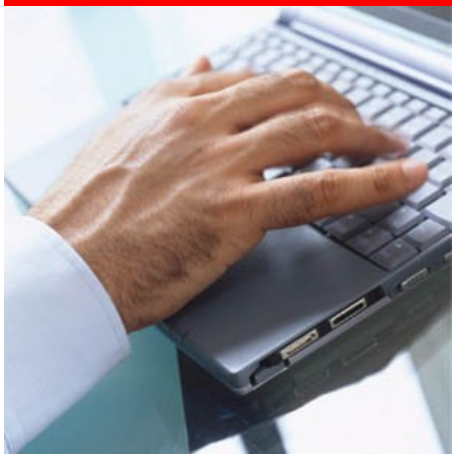

ORACLE®

ORACLE®

T10 Data Integrity Feature (Logical Block Guarding)

Martin K. Petersen
Software Developer, Linux Engineering

Topics



- Common Data Integrity Errors
- T10 Data Integrity Feature
- SCSI Layer Changes
- Block Layer Changes
- Performance Implications
- Discussion

Common Data Integrity Errors

- Misdirected writes
- Writing incorrect data
- On-the-wire corruption
- This actually happens in the field! Really!
- Allowing the storage device to verify data integrity before clobbering potentially good sectors
- Oracle's HARD

T10 Data Integrity Feature (DIF)

- Originally proposed by IBM
- Logical Block Guarding is one component of DIF
- SBC-3 / SPC-4
- 520 byte sectors with a twist
- 8 bytes of protection data per sector
- GUARD tag : CRC
- REFERENCE tag : Typically LBA
- APPLICATION tag : User defined content

T10 DIF – Tags

- GUARD tag (Logical Block Guarding):
  - 16-bit CRC covering the hardware sector
  - Regardless of sector size
  - 4096 KB sectors appear only to gain momentum in lower end (SATA)
- REFERENCE tag (Misdirected writes):
  - 4 bytes – depend on protection type
  - For Type 1 protection, REF tag contains lower 32 bits of LBA
  - For Type 2 protection, REF tag has to match LBA in CDB + N
  - Wraps at 2TB with 512 byte sectors, 16TB with 4KB

T10 DIF – Tags continued

- APPLICATION tag (Up for grabs):
  - 2 bytes per sector
  - Ownership negotiated with target
  - How do we provide this in a sensible way?
  - Per sector or per I/O?
  - Use it to flag metadata vs. data?
  - Ideas?

T10 DIF – Device Protection Types

- Type 0:
  - No checking but target device must generate on WRITE
- Type 1:
  - GUARD + REF checking (LBA)
- Type 2:
  - GUARD + REF checking (Extended Indirect LBA)
  - READ(32)/WRITE(32) only
- Type 3:
  - GUARD tag

T10 DIF – Device Capabilities

- Device can support one or more protection types
- Target can only be formatted with one protection type at a time
- RDPROTECT/WRPROTECT/VRPROTECT must match target format somewhat
- READ(32)/WRITE(32) feature special DIF knobs
- APP tag ownership/verification

T10 DIF – Host Board Adapters

- DIF is a standard for communication between initiator and target
- Some HBAs will likely use DIF transparently to OS:
  - INQUIRY/READ_CAPACITY(16) mangling
- Some may allow getting protection data from OS:
  - Allowing OS to submit a buffer with protection data included
  - Tag validity mask
- Some may allow DMA of protection data to OS:
  - Allowing OS to retrieve tags, including APP tag

T10 DIF – Protection Capabilities

- Protect all the way from filesystem to disk
- Which tags to supply are optional:
  - mount -o reference_tag
  - mount -o guard_tag
- If HBA is capable we can even protect path between OS and HBA with legacy storage devices
- Maybe even support DIF on legacy disks as long as they have 520 byte sector support (Academic Exercise)

SCSI Layer Changes

- Not very intrusive, except for sd.c CDB creation
- Error handling adapted to handle DIF-specific Additional Sense Codes + Qualifiers
- Distinguishes between HBA and target verification failures
- scsi_host mask to set HBA capabilities
- scsi_disk field to identify protection format

Block Layer Changes

- Propose a callback function which will calculate CRC and set APP + REF tags on a bio according to a tag mask
- bio_prot is a list of bio_vecs, mirroring the data vector
- “Protect this BIO if you can”
- Not SCSI-specific
- Filesystem doesn't have to be device capability aware

Block Layer Changes

- Will even work in case of RAID1 consisting of DIF and legacy disks
- But not with different sector sizes
- Merging of requests with mismatched bio_prot
- Ideas:
  - Need a way to communicate APP tag storage capability
  - Add a BH_Protect (BH_Integrity?) flag to buffer_head?
- Virtualization

Performance Implications

- CRC is somewhat expensive. 200-300 MB/s on a modern CPU
- Looking into ways to optimize
- SSE4 will have a CRC instruction (any poly)
- Protection data: 4KB page of protection data per 256KB of I/O