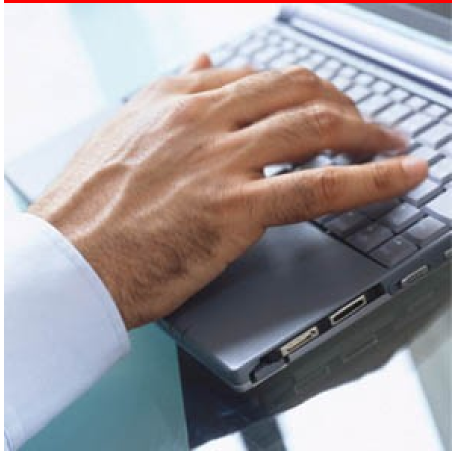
ORACLE®

Data Integrity Infrastructure for Block I/O

Martin K. Petersen
Software Developer, Linux Engineering

Topics



- Data Corruption
- Industry Update (T10/T13, DIX, SNIA)
- Linux Data Integrity Infrastructure
- Future Work / Discussion

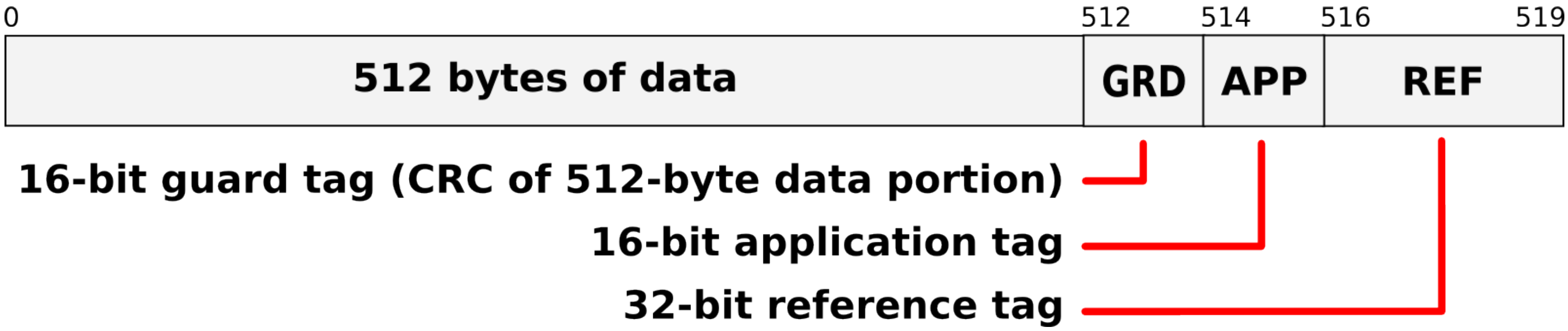
Data Corruption

- Tendency to focus on corruption while data is at rest
    - Media defects
    - Head misses
- However, corruption can happen while data is in flight
    - Modern transports like FC and SAS have CRC on the wire
    - Which leaves library / kernel / firmware errors
    - Bad buffer pointers
    - Missing or misdirected writes
- Industry demand for end to end checksumming
    - Oracle HARD is widely deployed
    - Other databases and mission-critical business apps
    - Nearline/archival storage wants belt and suspenders

Data Corruption - HARD/DIF/EPP

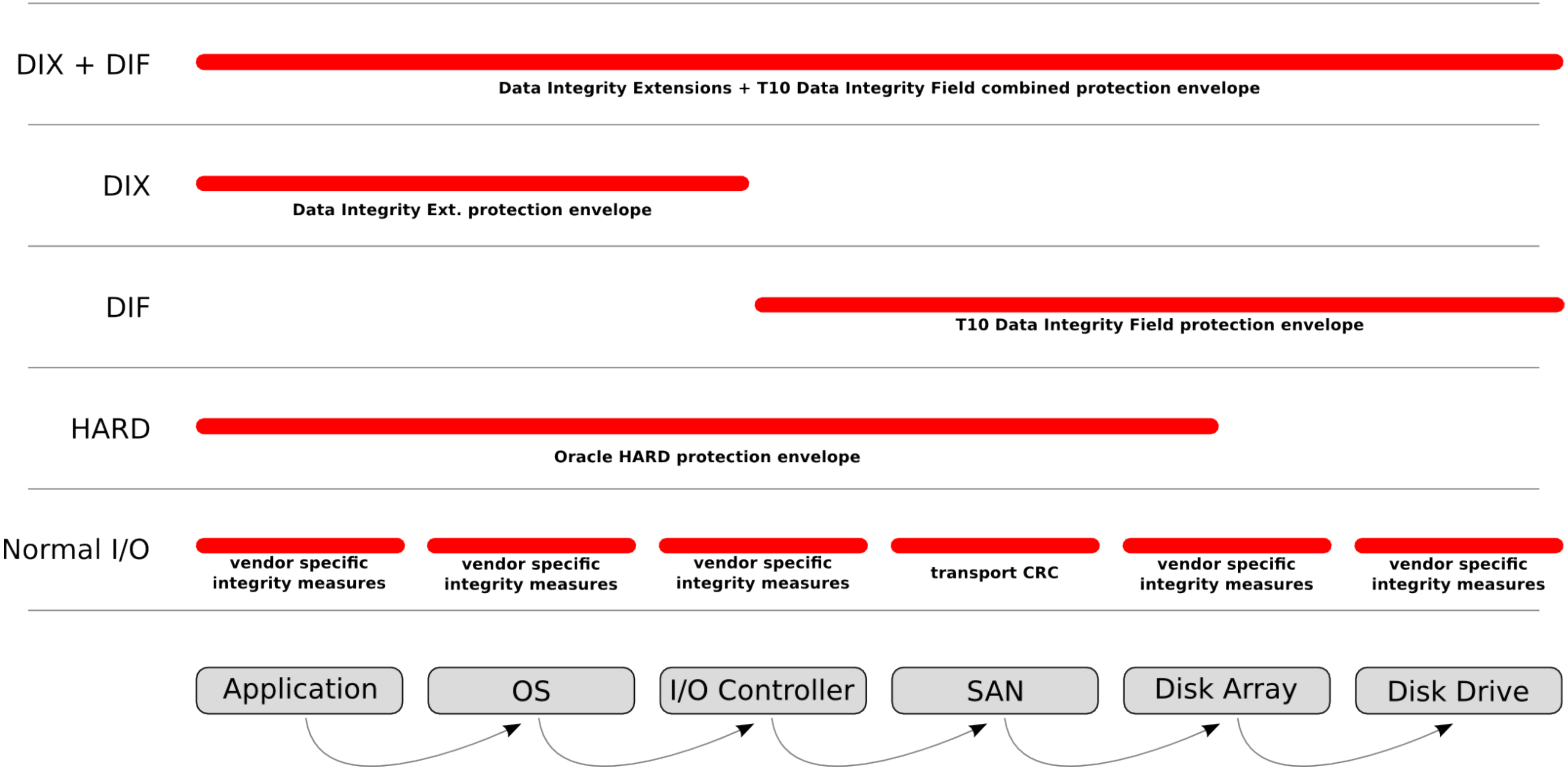
- Orthogonal to logical block checksumming
  - We still love you, btrfs!
  - Logical block checksumming is detected at READ time
  - ... which could be months later
  - Redundant copy may also be bad if buffer was incorrect
- This is about:
  - Proactively preventing bad data from being stored on disk
  - ... and finding out before the original buffer is erased from memory
  - Plus using the integrity metadata for forensics when logical block checksumming fails
- It's an insurance policy. Must be cheap.

T10 Data Integrity Feature (DIF)



- Between initiator and target
- IMD interleaved with data sectors on the wire
- Three protection schemes
  - All have guard tag defined
  - Type 1 reference tag is lower 32-bits of target sector
  - Type 2 reference tag is seeded in 32-byte CDB
- SATA T13/EPP uses same tuple format
- SSC tape proposal is different (guard only)

Data Integrity Extensions



DIX + DIF

Data Integrity Extensions + T10 Data Integrity Field combined protection envelope

DIX

Data Integrity Ext. protection envelope

DIF

T10 Data Integrity Field protection envelope

HARD

Oracle HARD protection envelope

Normal I/O

vendor specific integrity measures

vendor specific integrity measures

vendor specific integrity measures

transport CRC

vendor specific integrity measures

vendor specific integrity measures

Application

OS

I/O Controller

SAN

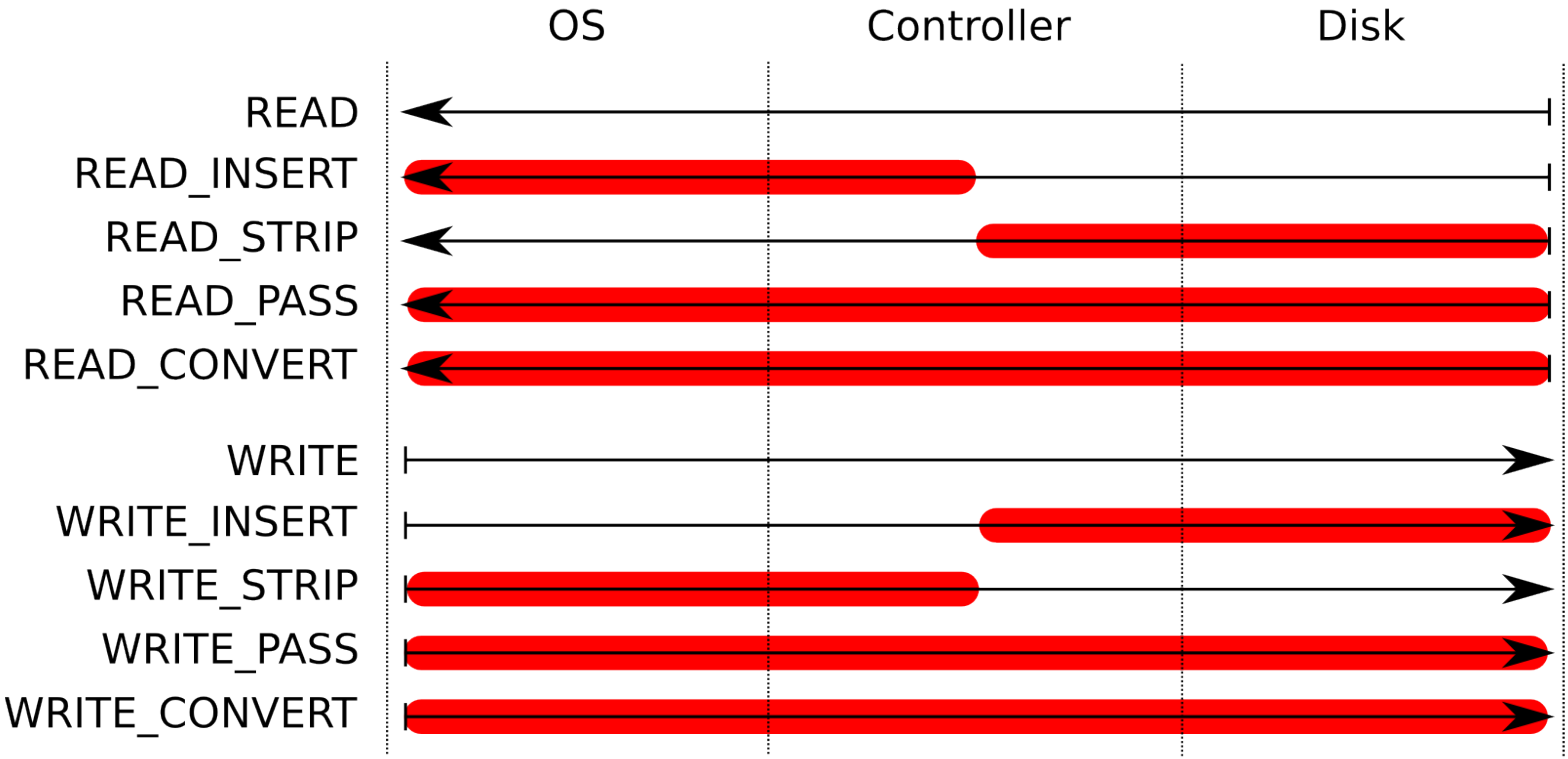
Disk Array

Disk Drive

Data Integrity Extensions

- Separate protection scatter-gather list
  - 520-byte sectors are inconvenient for the OS
  - A <512, 8, 512, 8, 512, 8, ...> scatterlist is also crappy
- DIF tuple endianness
  - Application tag must be portable across little- and big-endian systems
- Checksum conversion
  - CRC16 is somewhat slow to calculate
  - IP checksum is cheap
  - Strength is in data and integrity metadata buffer separation
  - CRC32 in Nehalem
  - Extra tags / protection schemes

DIX Operations



T10 DIF + Data Integrity Extensions

- Proof of concept last summer
  - Oracle DB, Linux 2.6.18, Emulex HBA, LSI array, Seagate drives
  - Error injection and recovery
- Product availability
  - Hardware shipping, firmware TBA
  - Emulex, LSI, Seagate, Hitachi

SNIA Data Integrity Technical WG

- Provisional TWG
- Aims to broaden participation
- Aims to standardize data integrity terminology
  - Think RAID levels
- Aims to standardize OS-agnostic API and/or common methods for applications to interact with integrity metadata
- Companies at first face 2 face
  - Emulex, Oracle, LSI, Seagate, Qlogic, Brocade, EMC, PMC Sierra, HP, Teradata, IBM, Sun, Microsoft, Symantec

What Is Now?

- SNIA is obviously a long-term effort
- “Verbatim” DIF exchange via DIX is pretty much good to go
- Linux infrastructure ready from block layer down
- Aiming for 2.6.26
- SCSI changes depend on block ditto

Linux Block Layer Changes

- `struct bio`
  - Integrity `bio_vec` + housekeeping hanging off of `bio`
  - Submitter can attach it
  - Or block layer can auto-generate on WRITE
  - Block layer can verify on READ
  - Integrity metadata opaque to block layer
- `struct block_device`
  - Has an integrity profile that gets registered by ULD
  - Layered devices must ensure all subdevices have same profile
- `struct request`
  - A few merging constraints
  - IMD ordering is important

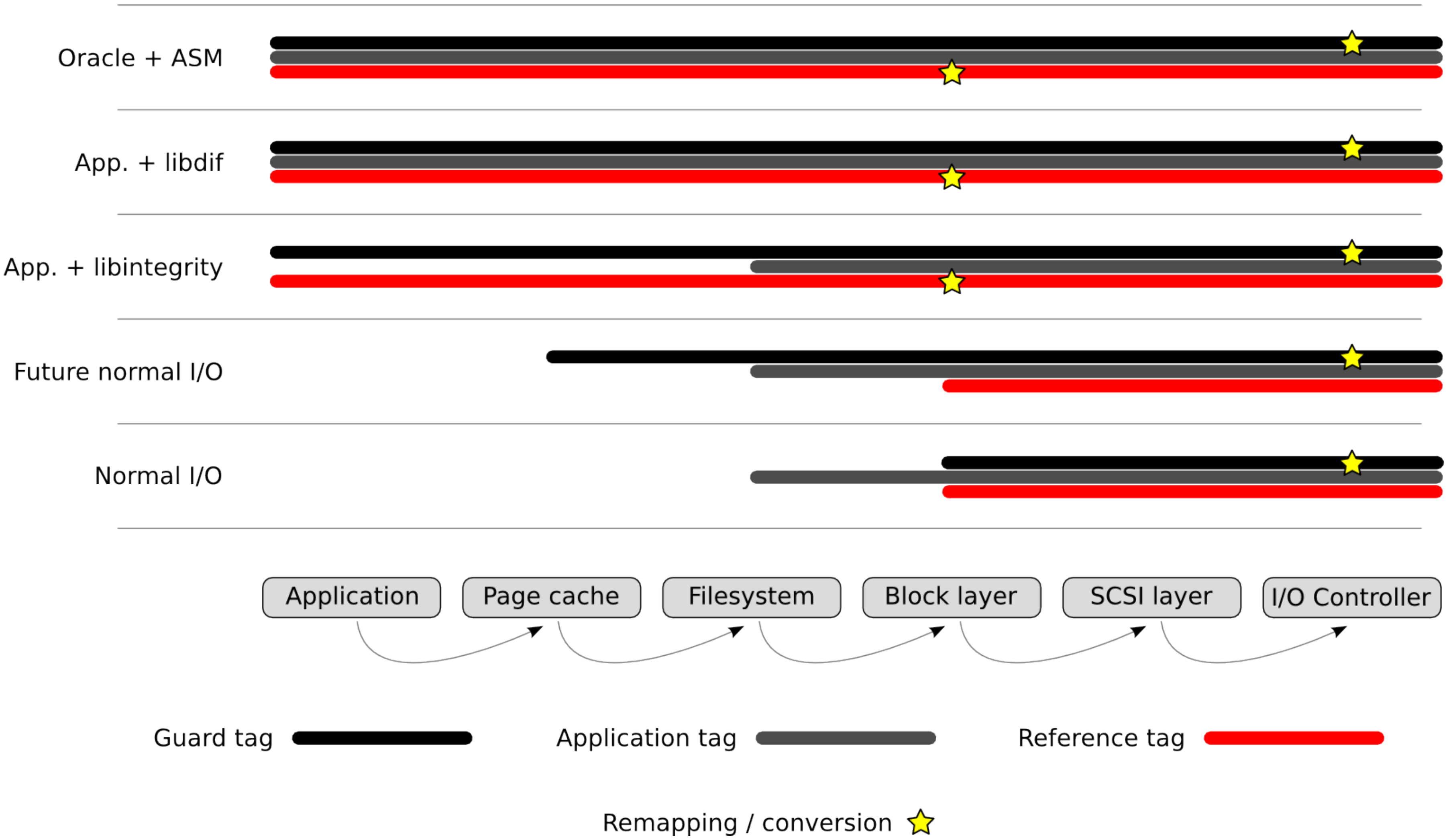
SCSI Layer Changes

- Mid level
  - `INQUIRY` and `READ CAPACITY(16)` during scan
  - Extra `scsi_data_buffer` in `scsi_cmnd`
  - Integrity scatter-gather mapping
- `sd.c`
  - CDB prep
  - A few knobs that HBA drivers can use to select DIX operation
  - Block integrity profile registration

Future Work / Discussion

- Filesystem / page cache interface
  - Where to pin? `address_space`? `struct page`?
  - FS application tag usage
- Userland API requirements:
  - Explicit
    - `mkfs`/`fsck` accessing DIF on block device directly
  - Opaque
    - “protect this buffer”
  - Transparent
    - standard `read()`/`write()` style calls
    - `mmap()` => bonghit bonanza

Application / OS Challenges



More Info

- http://oss.oracle.com/projects/data-integrity/
  - Documentation
  - DIX specification
  - Patches
  - Source repository