Storage Systems

NPTEL Course
Jan 2012
(Lecture 27)

K. Gopinath
Indian Institute of Science

Reliability Data for Future Large Systems

	'12	'15	'18	'20
Max Nodes (M)	0.1	0.1	1	1
Max Concurrency (M)	1	10	100	1000
Memory (DRAM) (PB)	4	10	30	60
MTTA Interrupt (days)	1	1	0.5	0.25
Storage Application Visible interrupt (days)	20	18	16	14

Storage Reliability Data

BER

Consumer SATA 1.0E-14

Enterprise SATA 1.0E-15

Enterprise SAS/FC 1.0E-16

LTO 1.0E-17

Undetectable errors (channel error rate 1.0E-12 for all)

Ethernet 1.0E-21 estimated

SATA 1.0E-17

Fibre Channel 1.0E-12 1.0E-21

SAS 1.0E-21

FC/SAS with T10 PI 1.0E-28

Data Corruption

Errors in the magnetic media

- On disks, Reed-Solomon codes used

Errors induced through drive firmware

- eg. *phantom write*

Corruption when the data travels

- Protect using CRC, IP checksums

Software errors

Handling errors

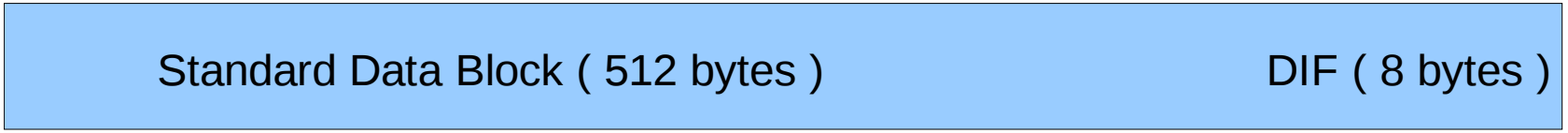
- Enterprise servers use RAID to protect against drive failure
- System busses have protective measures such as parity
- Modern file systems such as *btrfs, zfs* have checksum on data as well as metadata
- But no standardized way to ensure “end-to-end” integrity till recently
  - Data Integrity Field (DIF): for storage stack
  - Data Integrity eXtension (DIX): above storage stack

Standards-based Protection

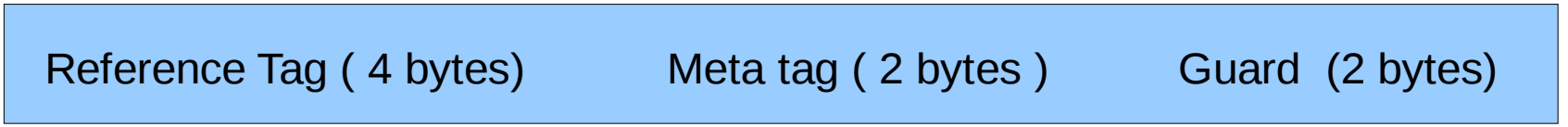
End-to-End Data Protection std

- approved by T10 Technical Committee of INCITS/ANSI

An extension to SCSI block device protocol known as *Data Integrity Field (DIF)*



DIF: an *8 byte* header for the standard *512 byte* sector



Guard tag: a 16-bit CRC of the sector data.
Meta tag (Application tag): a 16-bit value that can be used by OS
Reference tag: a 32-bit number used to ensure individual sectors written into right physical sector (depending upon type the drive supports)

DIF in operation

- HBA
  - calculates a checksum value for the data block on writes
  - stores it in the DIF
- Storage device
  - confirms checksum on receipt
  - stores the data plus checksum
- On a read, checksum
  - checked by the storage device
  - also by the receiving HBA

DIF enough?

T10 standardizes 520 bytes/sector for SCSI drives with DIF format

- prevents content corruption & misplacement errors
- protects path between HBA & storage Device

But does not provide integrity for software stack (such as filesystem), where errors like bad buffer pointers, misdirected writes occur

Hence need also to provide integrity in filesystem over DIF.

ZFS error correction

- ZFS uses end-to-end checksums
- Checksum not stored with the data block, but in the pointer to the block
- All checksums done in server memory, so catches
  - Phantom writes, where the write is “null”
  - Misdirected reads or writes, where disk accesses the wrong block
  - DMA parity errors between a storage array and server memory or from the driver, since checksum validates data inside array
  - Driver errors, where data stored in the wrong kernel buffer
  - Accidental overwrites, such as swapping to a live file system
- Cost: approx 1 GHz of CPU time to process 500 MB/sec I/O
  - 1 cycle of CPU for 4 bits of I/O!
- Not a standard!

Data Integrity Extension (DIX)

DIX defines a set of interfaces that host adapters must provide in order to exchange integrity metadata with the host operating system

- application calculates checksum and passes it to the HBA, to be appended to the 512 byte data block.
- provides a full end-to-end data integrity check

To provide End-to-End protection

- controller has to be DIX capable
- storage device DIF capable

General Method for Checksumming Above HBA

- use SCSI DIF/DIX to ensure integrity of data in filesystem using the *application tag*
- use algebraic signatures to ensure integrity of both data and meta-data
- useful property: if can calculate algebraic signature of the whole block given algebraic signatures of sub-blocks

Galois Field

- Galois field is a field with finite number of elements represented *GF(n)*
- Number of elements in *GF* is p^n, where *p* is prime
  - Here, only GF(2^n), specifically GF(2^{16}).
- Elements in form of polynomials, binary strings etc
  - $x^4 + x^2 + 1$ <=> 10101
- Addition, Subtraction equiv to XOR, hence commutative
  - Similar defs for Multiplication, Division
- An element *α* in the field is irreducible if it cannot be expressed as product of non trivial factors
- ord (*α*) of a non-zero element *α* is the smallest non zero exponent *i,* such that $\alpha^i=1$
- An element *α* is a *primitive element* if ord(*α)=s-1,* where *s* is the size of GF

Algebraic Signatures

An algebraic signature of string of symbols $x_1, x_2 \ldots x_N$ is defined by:

$$sig_\alpha (x_1 , x_2 , x_3 \ldots x_N) = \sum_{v=1..N} x_v . \alpha^v$$

Important properties of algebraic signatures

- If block length $l < \text{ord}(\alpha)$, where α is a primitive element and every possible block content is equally likely, then the signatures sig_α of two different sectors collide with a probability of 2^{-f},where f is the length of each symbol
- $sig_\alpha (x_1 + x_2) = sig_\alpha (x_1) + sig_\alpha (x_2)$

Can use algebraic signatures on sub-blocks and blocks in a FS to ensure integrity

- signatures summarizes protection information over longer sequences of information units such as blocks
- eg. for a *block* B_n the application tag split into 2 parts
  - IntegritySubBlk_n stores the integrity metadata of *block* B_n
    - Stores integrity data of each subblock
  - $\text{IntegrityBlk}_{n+1}$ holds the integrity metadata of *block* B_{n+1} (summary)

Conclusions

- Error Management will become central in future large designs
- Need support across various components of the storage and appl stack