RAID6L: A Log-Assisted RAID6 Storage Architecture with Improved Write Performance

Chao Jin*, Dan Feng*, Hong Jiang†, Lei Tian*†

*Huazhong University of Science and Technology

*Wuhan National Lab for Optoelectronics

†University of Nebraska-Lincoln

Background

- Reed-Solomon Coded RAID6 Architecture
  - Two Parities per Parity Stripe: P & Q
  - P is computed by *XOR* Parity
  - Q is computed by *Finite Field* Arithmetic
- RAID6 Write Performance
  - P&Q Parity Update upon Each Data Update
  - High Write Penalty and Low Write Performance

Contributions

- Generalized Parity Logging
  - Improving Write Performance of RAID with Logging Technique
  - Original Parity Logging: Applicable to XOR based RAID (e.g., RAID5)
  - Generalized Parity Logging: Applicable to Any Form of RAID (e.g., Reed-Solomon Coded RAID6)
- RAID6L
  - New RAID6 Architecture based on Logging
  - Improved Write Performance
  - Advantageous over Generalized Parity Logging
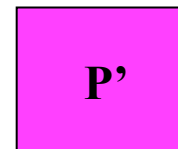
Original Parity Logging

RAID 5

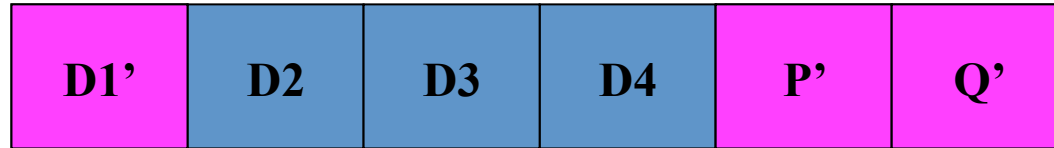| D1'	D2	D3	D4	P'

Parity Logging

Log Disk

D1'	⊠	D3	D4	P

D1 ⊕ D1'

Data Block Recovered!
Data Block Failed!

P'

Generalized Parity Logging
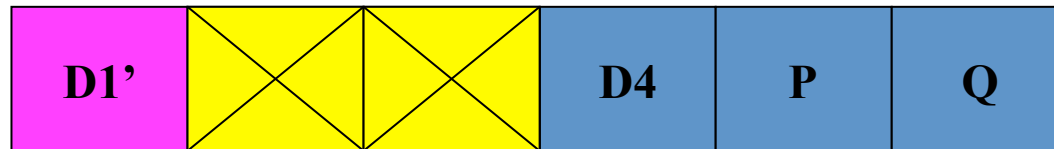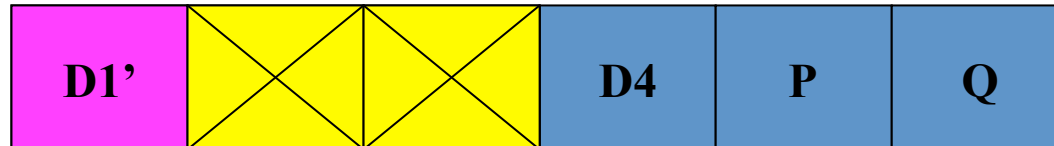
RAID6L Architectural Overview

Stores the metadata of the log disk: log address of the original/ current value of data blocks

Workload

Hash List

Query

Update

Request Handle

Parity Re-synchronize

Failure Recover

State Monitor

Request Dispatch

Data Disks

Parity Disks

Log Disk

Monitor the State of RAID6L:
Accelerate State: **Stop Parity Update and Start Logging;**
Transitional State: **Parity Resynchronization ;**
Normal State: **Regular RAID Procedure**

RAID6L: Write Procedure and Failure Recovery

- ## Utilization of Workload Locality
  - RAID6L: Yes (Save Pre-read Operations!)
  - Generalized parity Logging: No

Parity Stripe

A0	B0	C0	D0	E0	P0	Q0

A1	B1	C0	D0	E0	P0	Q0

A2	B2	C0	D0	E0	P0	Q0

Parity Logging

A0⊕A1
B0⊕B1
A1⊕A2
B1⊕B2

RAID6L

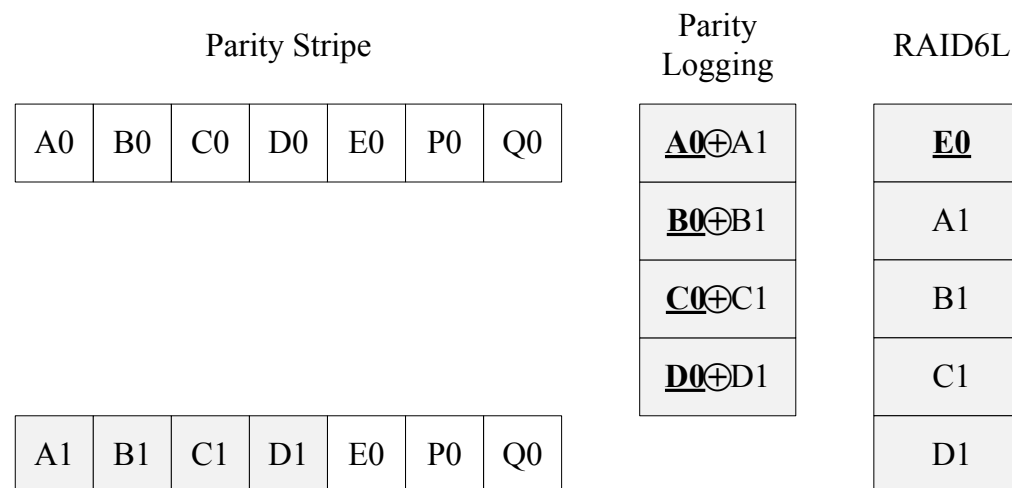| **A0** |
|--------|
| **B0** |
| A1 |
| B1 |
| A2 |
| B2 |
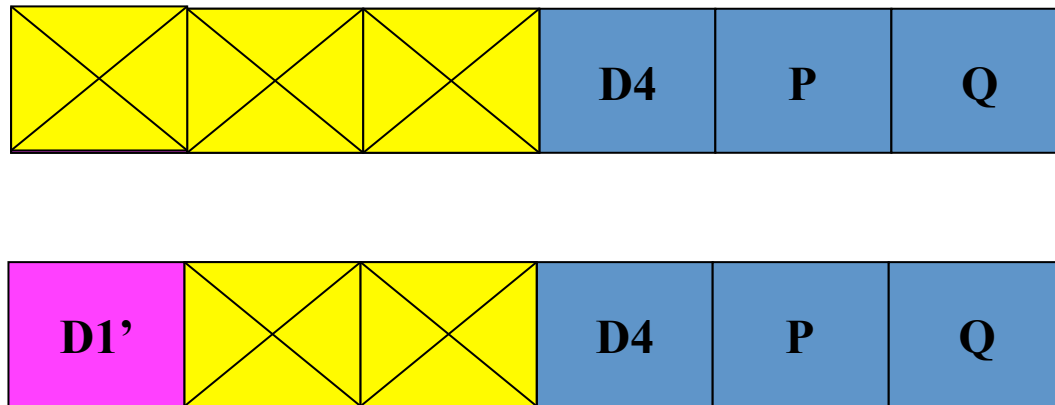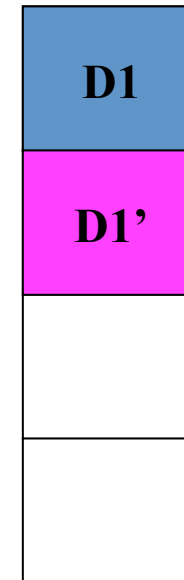
Comparison of RAID6L and Generalized Parity Logging (2/3)

- ## Large Write Optimization
  - RAID6L: Choose RMW or RCW Dynamically (Save Pre-read Operations!)
  - Generalized parity Logging: RMW Only

Parity Stripe

A0	B0	C0	D0	E0	P0	Q0

A1	B1	C1	D1	E0	P0	Q0

Parity Logging

A0$\oplus$A1
B0$\oplus$B1
C0$\oplus$C1
D0$\oplus$D1

RAID6L

E0
A1
B1
C1
D1

Comparison of RAID6L and Generalized Parity Logging (3/3)

- ## Data Recovery Efficiency
  - ### RAID6L
    - Updated Data Blocks Recovered by Directly Copying
    - Unupdated Data Blocks Recovered by Directly Copying or by RAID6 Parity Algorithm
    - Small Part of Log Records are Used
  - ### Generalized parity Logging
    - Compute Original Value of Survival Data, Then Compute the Original Value of Failed Data by RAID6 Parity Algorithm, Finally Compute the Current Value of Failed Data
    - All Log Records are Used

Thanks!

Email:

chjinhust@gmail.com