Ping Xie, Jian-zhong Huang, Er-wei Dai, Qiang Cao, Chang-sheng Xie, 2015. An efficient data layout scheme for better I/O balancing in RAID-6 storage systems. *Frontiers of Information Technology & Electronic Engineering*, **16**(5):335-345.
[doi:10.1631/FITEE.1400362]

An efficient data layout scheme for better I/O balancing in RAID-6 storage systems

Key words: RAID-6, Data availability, High performance, I/O balancing

Contact: Ping Xie
E-mail: xieping@qhnu.edu.cn
ORCID: http://orcid.org/0000-0001-9122-8534

Contents

1 Motivation and background

2 Uniform P-code

3 Property analysis

4 Performance evaluation

RAID-6

- The Storage Networking Industry Association (SNIA) defines RAID-6 as

Any form of RAID that can continue to execute read and write requests to all of a RAID array's virtual disks in the presence of any two concurrent disk failures.

Existing RAID-6 codes

Non-MDS codes

(lower storage efficiency)

- LDPC [DSN'04]
- WEAVER [FAST'05]
- HoVer [DSN'06]
- MEL [DSN'07]
- Flat-XOR [MSST'10]
- Code-M [DSN'10]
- …

MDS codes

(higher storage efficiency)

- Horizontal codes
  - Reed-Solomon
  - EVENODD [TC'95]
  - RDP [FAST'04]
  - Liberation [FAST'08]
  - …
- Vertical codes
  - X-Code [TIT'99]
  - P-Code [ICS'09]
  - …

Other codes (tolerate three or more disk failures): RSL and RL [TC'05], STAR [FAST'05], GRID [TOS'09], etc.

MDS codes

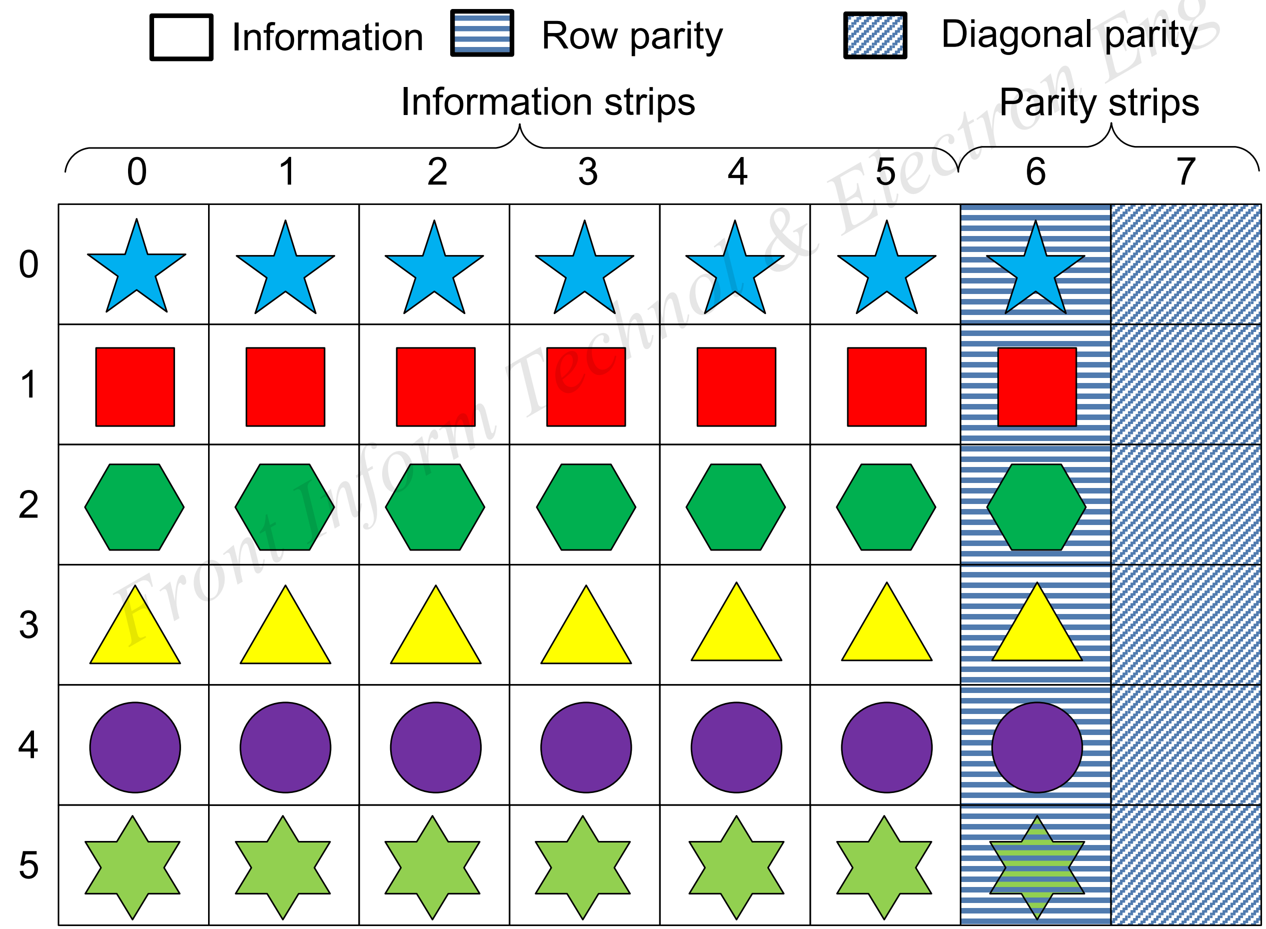
- **Horizontal codes**
  - Dedicated parity disks, such as EVENODD and RDP, which can become potential performance bottleneck in context of write-intensive
- **Vertical codes**
  - Parity symbols are dispersed over all disks, such as X-code and P-code
  - Achieving better I/O balancing relative to horizontal codes

MDS codes

- **Optimal storage efficiency**
  - Attain the Singleton bound
- **A typical MDS code is composed of *k*+2 disks array**
  - The first *k* disks store original information
  - The last two are used as parity disks
  - Parity chain length is *k*+1
- **The limitation of dedicated parity disks**
  - Corresponding parity blocks need to be updated for any write operation, thereby causing heavier workload on parity disks, which potentially degrades the access performance.

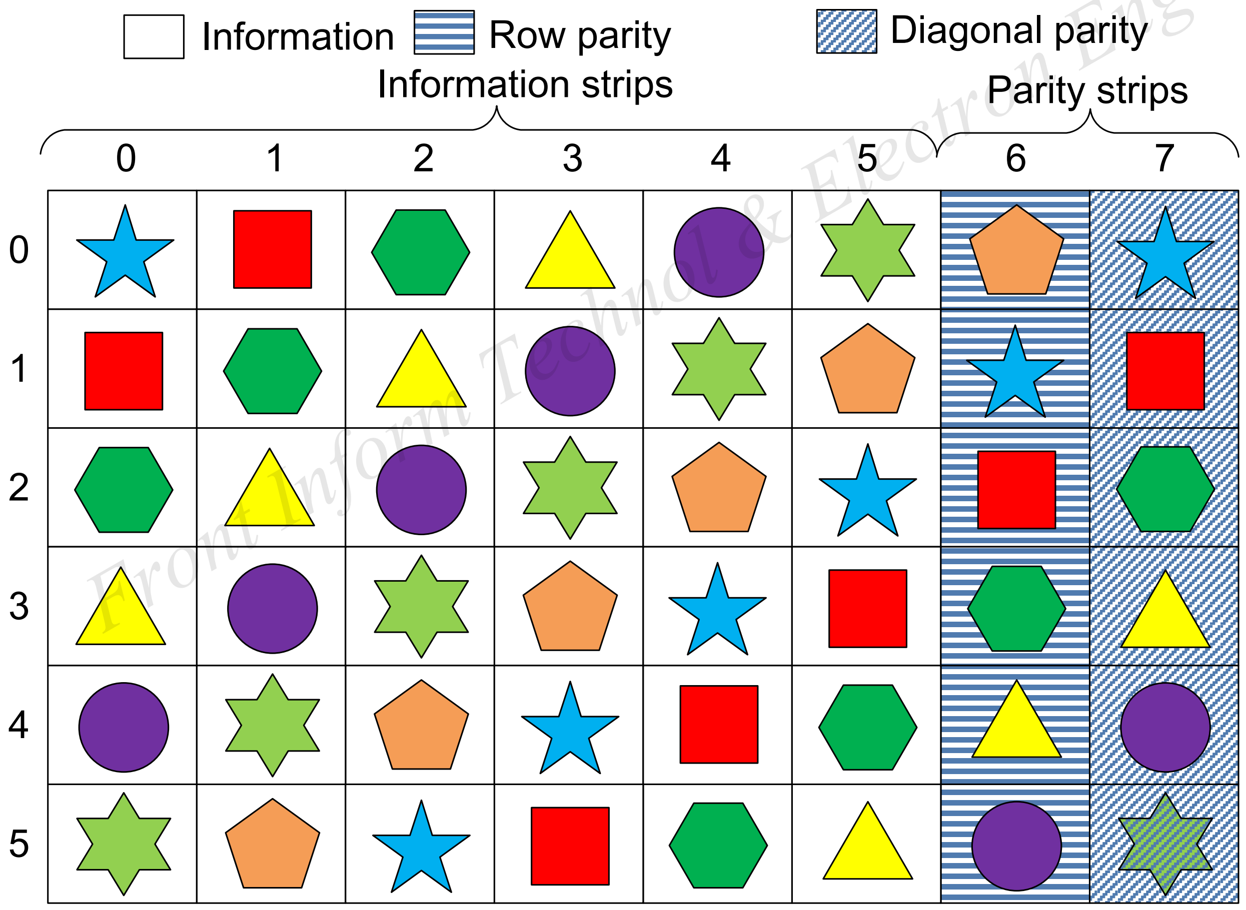
I/O balancing problem in RDP

❖ Horizontal parity layout (p=7)



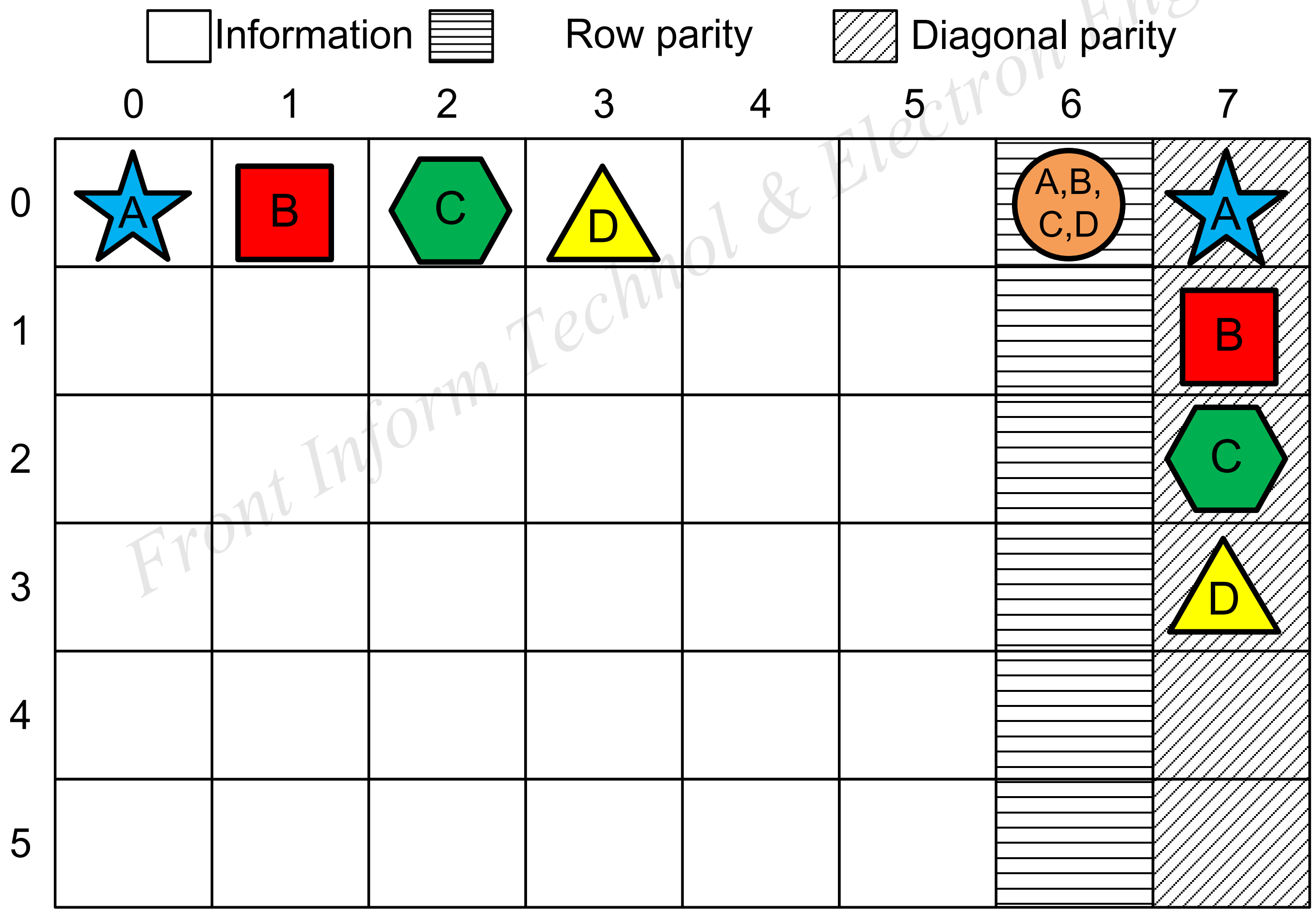
I/O balancing problem in RDP

Diagonal parity layout (p=7)



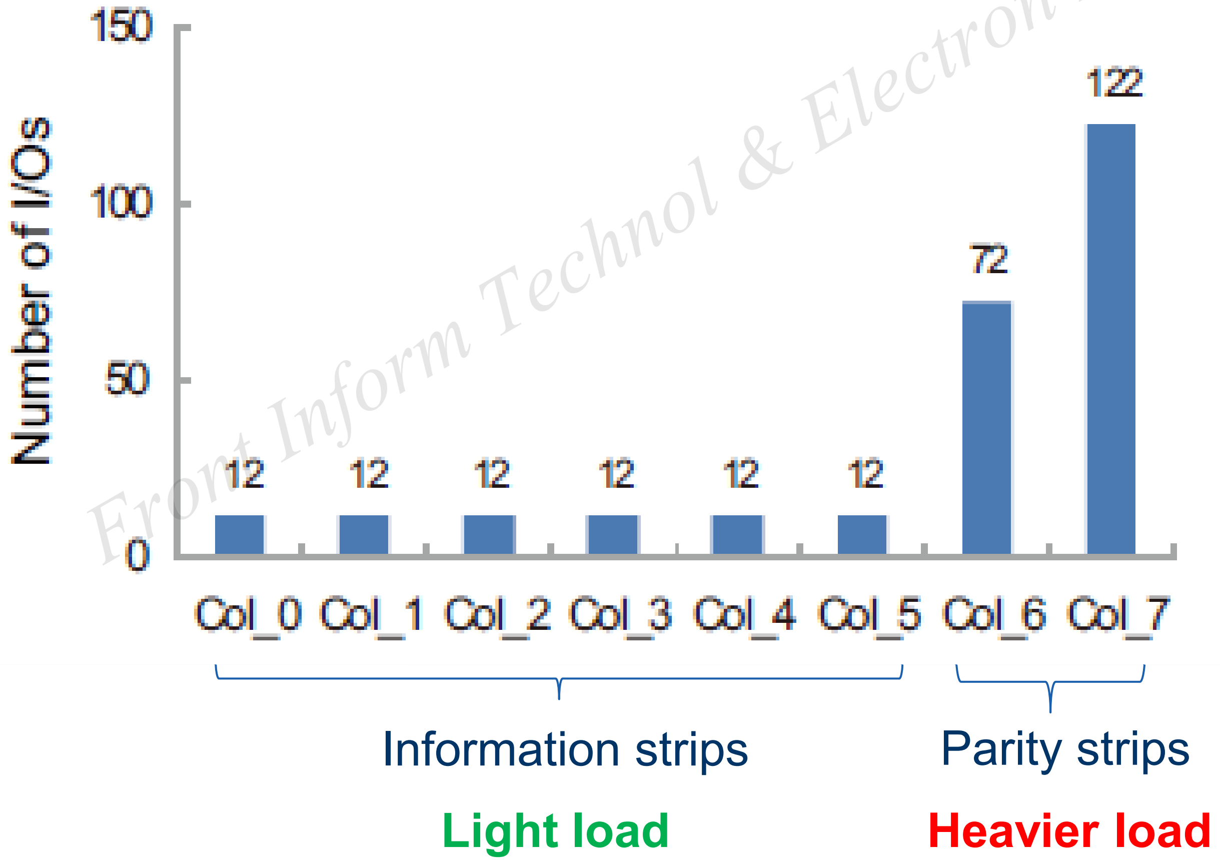
I/O balancing problem in RDP

- A partial stripe write to four continuous information symbols: A, B, C, and D



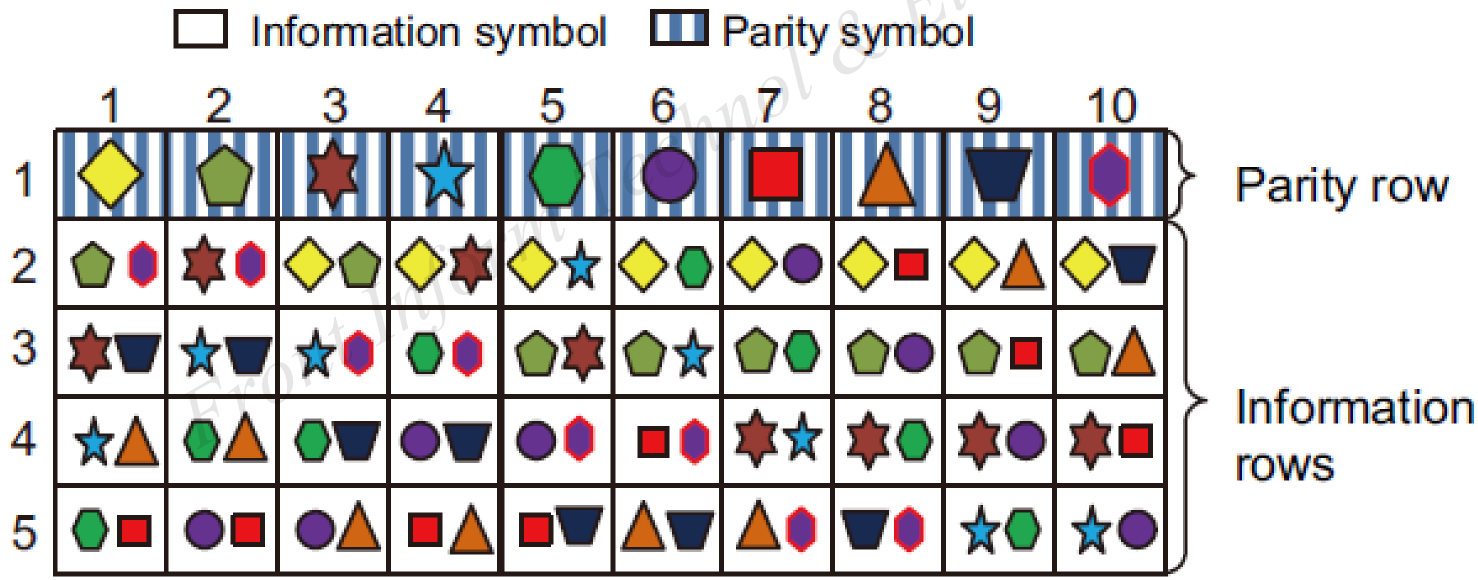
Unbalanced I/O in RDP

- Uniform single write to all information blocks



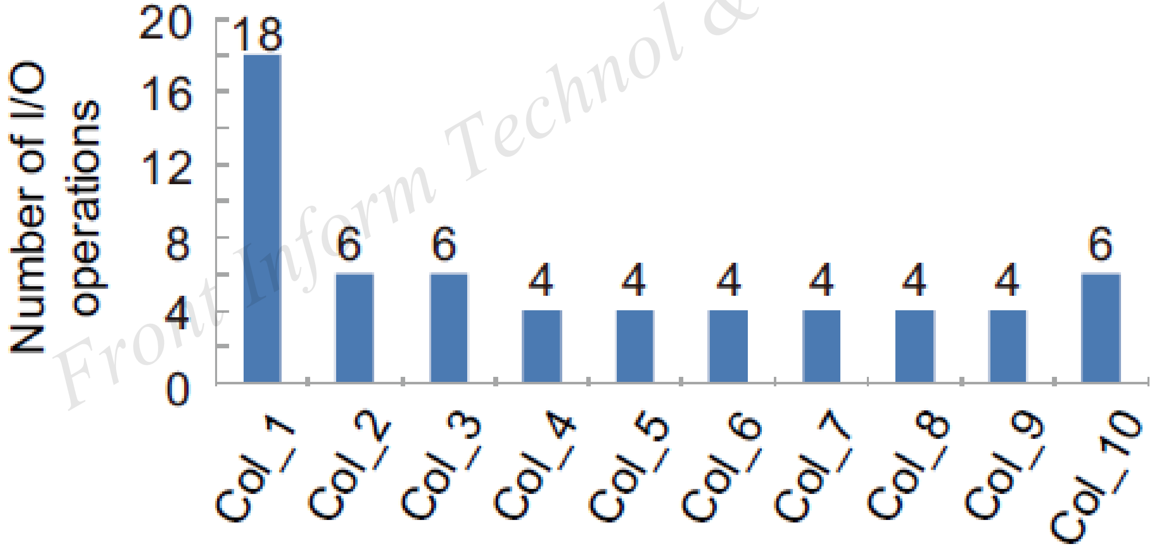
I/O balancing problem in P-code

- Vertical parity layout (p=11)



Unbalanced I/O in P-code

- Uniform single write to the second row (information row)



Motivation

- Uneven information symbol distribution among all disks in P-code
- Balancing the I/O distribution among all disks

Contents

1. Motivation and background
2. Uniform P-code
3. Property analysis
4. Performance evaluation

Uniform P-code

❖ Vertical parity layout (p=11)

Information symbol

Information symbol in a parity chain

Parity symbol

Parity symbol in a parity chain

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(2,10)	(3,10)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)
(3,9)	(4,9)	(4,10)	(5,10)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)
(4,8)	(5,8)	(5,9)	(6,9)	(6,10)	(7,10)	(3,4)	(3,5)	(3,6)	(3,7)
(5,7)	(6,7)	(6,8)	(7,8)	(7,9)	(8,9)	(8,10)	(9,10)	(4,5)	(4,6)

(a)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(2,10)	(6,7)	(4,10)	(1,3)	(7,9)	(8,9)	(1,6)	(3,5)	(4,5)	(2,8)
(3,9)	(4,9)	(6,8)	(5,10)	(1,4)	(7,10)	(2,5)	(2,6)	(1,8)	(3,7)
(4,8)	(3,10)	(5,9)	(6,9)	(2,3)	(1,5)	(8,10)	(1,7)	(2,7)	(4,6)
(5,7)	(5,8)	(1,2)	(7,8)	(6,10)	(2,4)	(3,4)	(9,10)	(3,6)	(1,9)

(b)

Fig. 4 Data layout comparison between the P-code (a) and uniform P-code (b) with a 10-disk array

❖ Moving methods (Algorithm 1)

Contents

1 Motivation and background

2 H-code

3 Property analysis

4 Performance evaluation

Properties of uniform P-code

- **Optimal storage efficiency**
  - MDS attains the Singleton bound
- **Optimal computational complexity**
- **Optimal update complexity**
  - Updating one information symbol only needs to update two parity symbols: *optimal update complexity*
- **Exhibit better I/O balancing and higher data availability compared to the P-code scheme**

Contents

1. Motivation and background
2. H-code
3. Property analysis
4. **Performance evaluation**

Comparison results

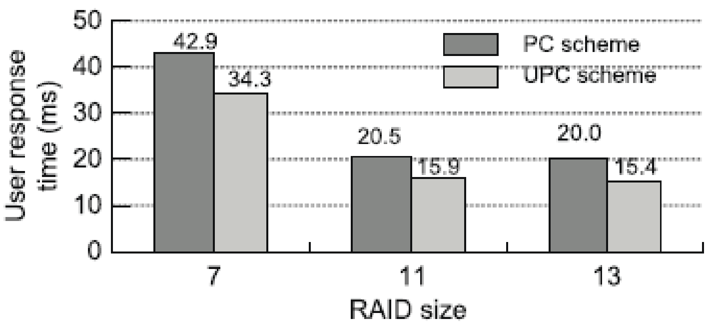


Fig. 5 Average user response time under different RAID sizes

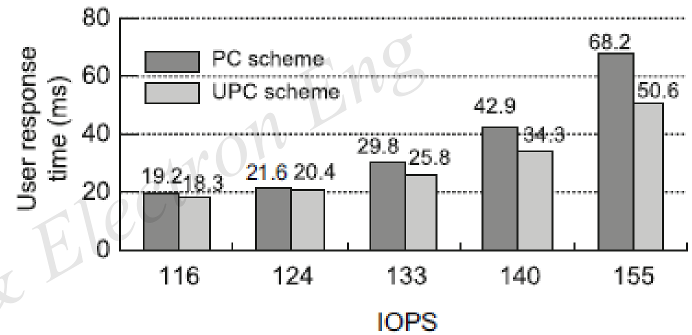


Fig. 6 Average user response time under different IOPSs ($p = 7$)

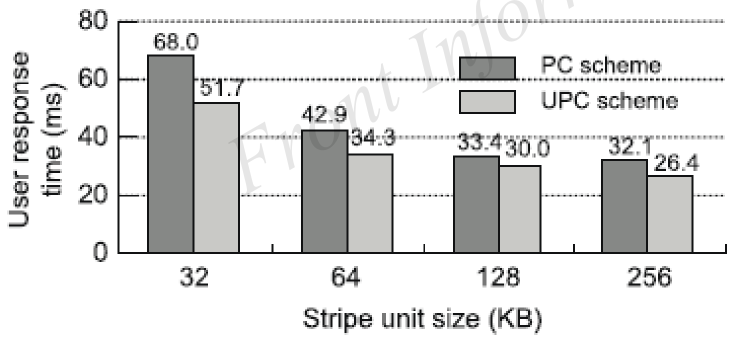


Fig. 7 Average user response time under different stripe unit sizes ($p = 7$)

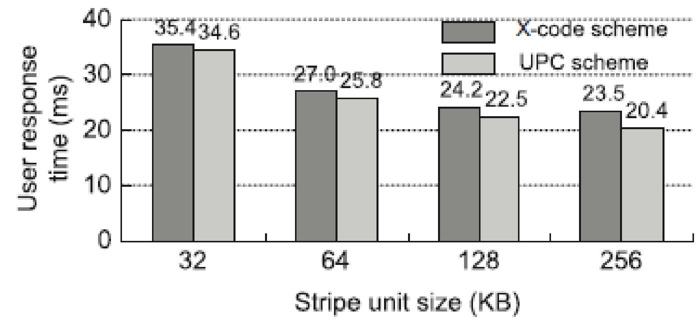


Fig. 8 The performance evaluation of UPC and X-code ($p = 7$)

Conclusions

- **This paper presents a novel code in RAID-6: uniform P-code**
  - Optimal storage efficiency, computational complexity, and update complexity
  - Improved write performance compared to existing P-code in RAID-6 storage system