The 35th International Conference on Massive Storage Systems and Technology (MSST 2019)

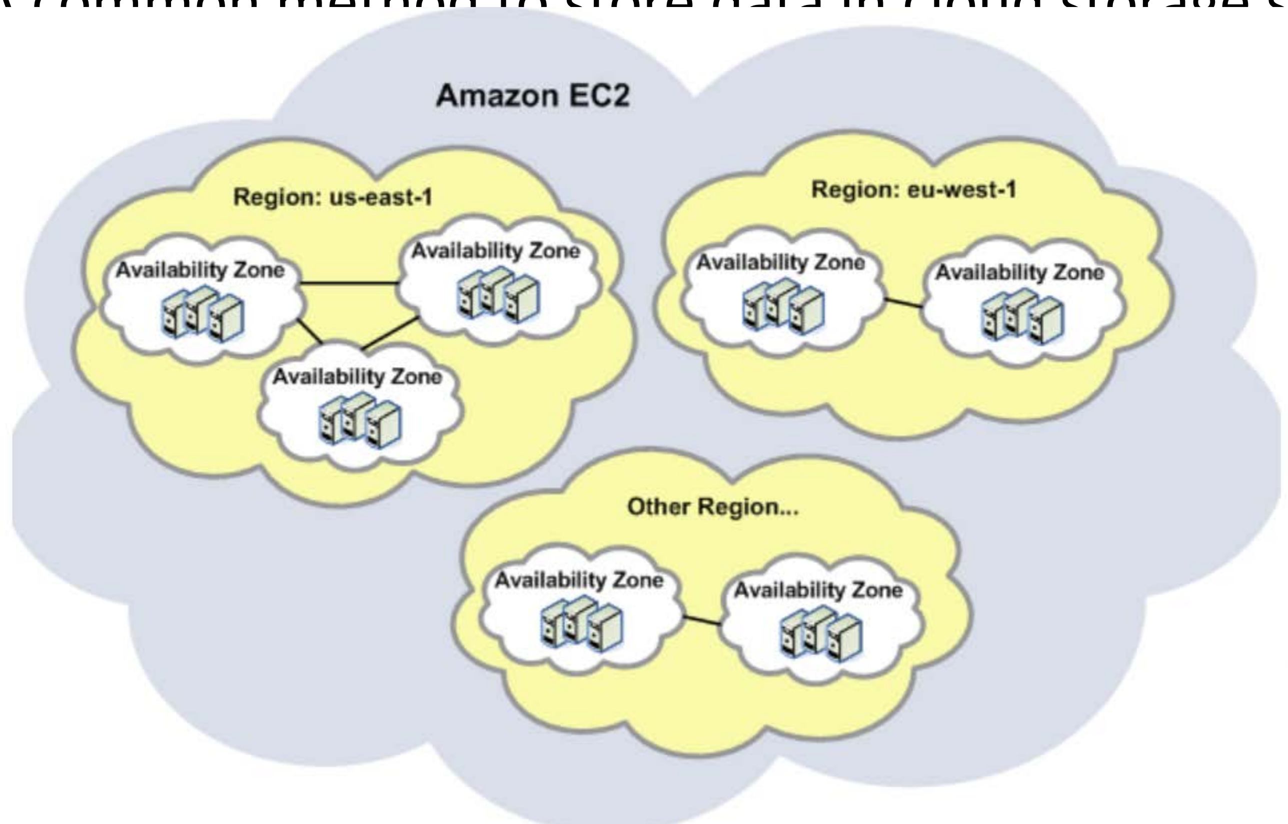
AZ-Code: An Efficient Availability Zone Level Erasure Code to Provide High Fault Tolerance in Cloud Storage Systems

Xin Xie, **Chentao Wu**, Junqing Gu, Han Qiu, Jie Li, Minyi Guo, Xubin He, Yuanyuan Dong, and Yafei Zhao

Content

Background

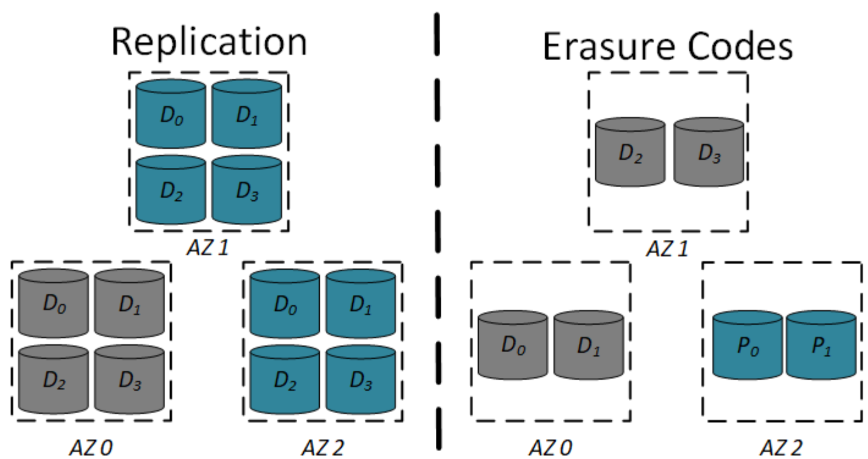
- Availability Zones
  - A common method to store data in cloud storage system
  - l … nd
  - A
  - E
- Dat
  - F
  - e
  - F



https://amazonaws-china.com/cn/

Erasure Codes

- Traditional replication strategies
  - High reliability & High storage cost
- Erasure Codes (EC)
  - High reliability & Low storage cost
  - 2 types:
    - XOR-based erasure codes (e.g. EVENODD [TC'95], STAR code [FAST'05], TIP-Code[DSN'2015])
    - RS-based erasure codes (e.g. RS code, LRC[ATC'2012][ISIT'15])

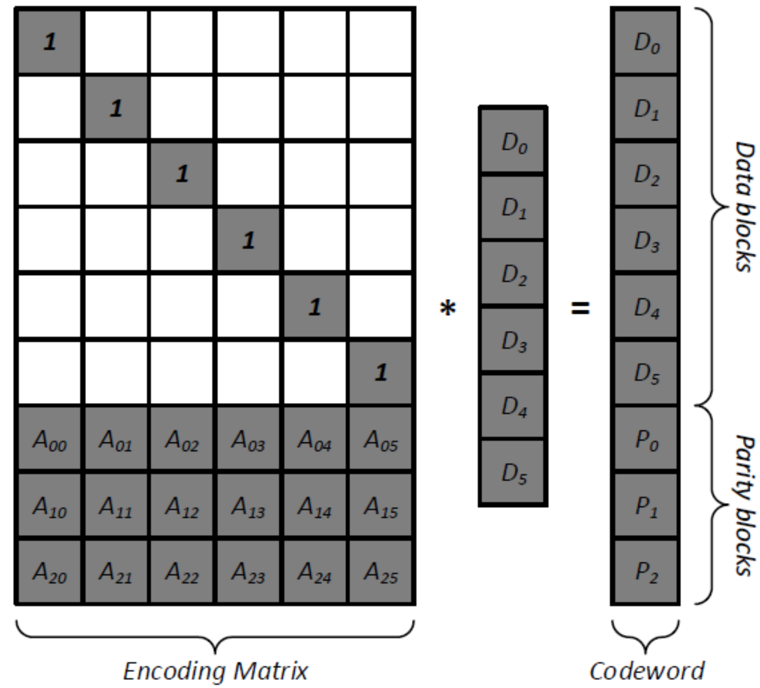


Requirements of Erasure Codes in Multiple AZs

- Requirements of Erasure Codes in Multiple AZs :
  - High Reliability
    - tolerate concurrent AZ level failures (e.g., lost a whole AZ)
  - High Storage Efficiency
    - make the storage efficiency as high as possible in AZ environments
  - Low Recovery Cost
    - Low computation cost
    - Low network bandwidth requirement
    - Small number of I/Os
  - High Scalability
    - provide fast scaling processes when scale up or scale out

Classic Erasure Codes in Cloud Storage Systems : RS Code

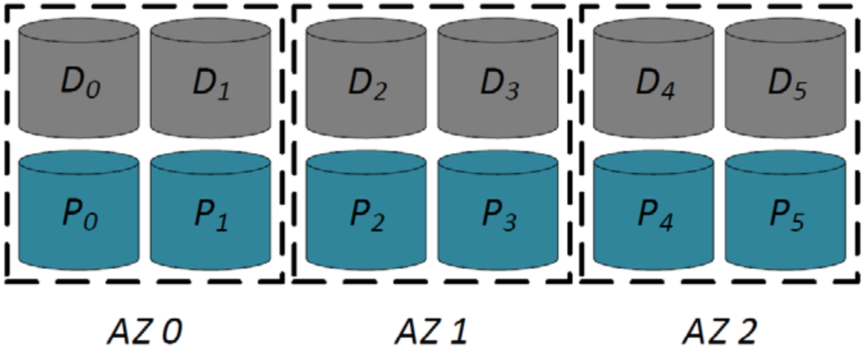
- Reed Solomon Code [Irving S. Reed Gustave Solomon 1960], usually represented as RS(k,r)
    - k: the number of data blocks.
    - r: the number of parity blocks.
    - n: the total number of blocks (n=k+r).
- All operations of RS code are based on Galois Field



The encoding process of RS(6,4)

RS Code in Multiple AZs

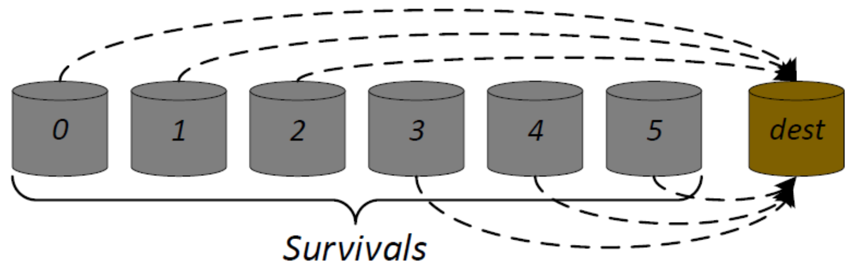
- RS Code can be applied in multiple AZs
  - All nodes are evenly distributed among all AZs.



RS(6,6) in 3 AZ

- Recovery process need many data nodes -----

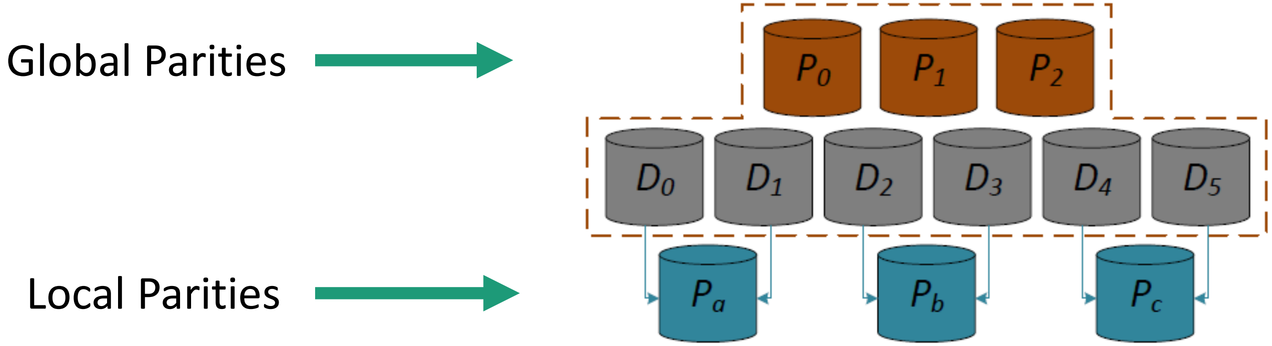
  high recovery cost



The decoding proess of RS(6,6)
(one failure node)

Classic Erasure Codes in Cloud Storage Systems :LRC [ATC'2012][ISIT'15]

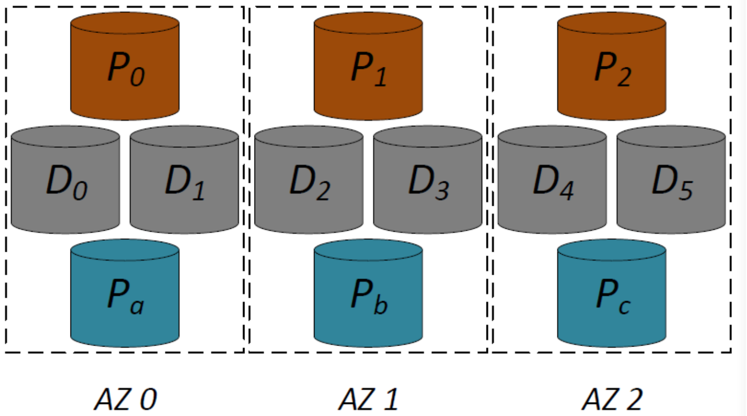
- LRC(k, z, r)
  - k: the number of data blocks.
  - z: the number of groups.
  - r: the number of parity blocks.
  - Applied in Windows Azure Storage and Facebook cloud
- Main idea: Use local parities, but increase storage cost



LRC(6,3,3)

LRC in Multiple AZs

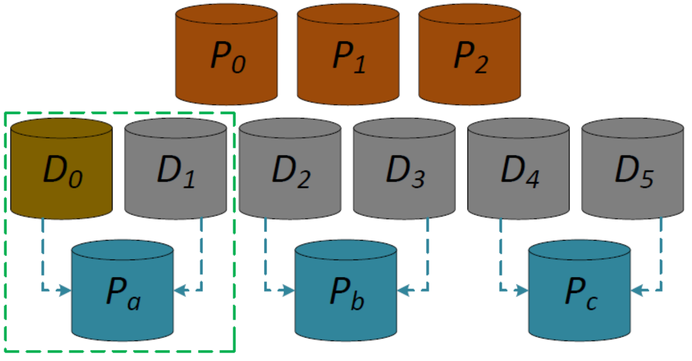
- LRC can be applied in multiple AZs
  - All nodes are evenly distributed among all AZs.



LRC(6,3,3) in 3AZ

- Recovery process need local data nodes -----

  lower recovery cost



The decoding proess of LRC(6,3,3) (one failure node)

Existing Erasure Codes in Cloud Storage Systems : Other Methods

- XOR-based Codes
  - Low computational complexity
  - Scalability issue
  - E.g. STAR [FAST'05], TIP-Code [DSN'15], GRID codes [ACM TOS'09], etc.
- MSR Code: one special case of regenerating code with the property of Maximum Distance Separable (MDS) codes
  - Low recovery cost
  - Very high computational complexity

Summary on Existing Erasure Codes in Multiple AZs

	Reliability	Scalability	Storage Efficiency	Recovery Cost	Computational Complexity
RS Code	✓	✓	✓	✗	✗
LRC Code	✓	✓	✗	✓	✗
MSR Code	✓	✗	✓	✓	✗
GRID Code	✓	✗	✓	✗	✓
AZ-Code	✓	✓	✓	✓	✓

Content

01 Background

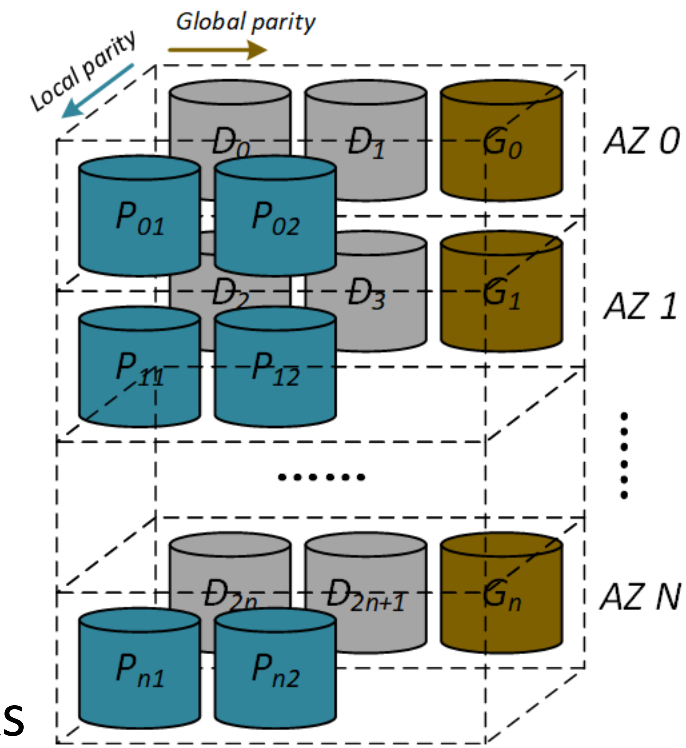
02 AZ-Code

03 Evaluation

04 Conclusion

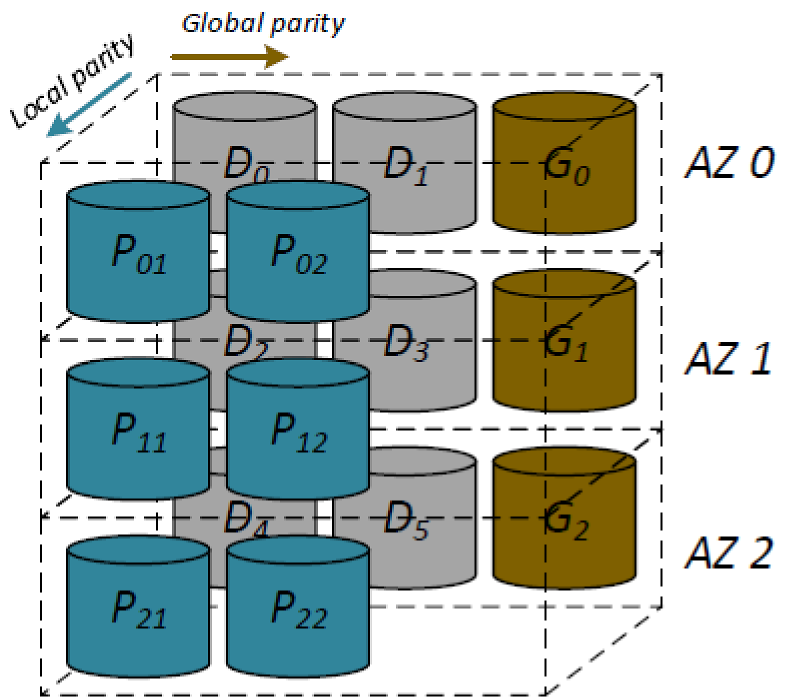
AZ-Code Overview

- Our idea: Combination of LRC and MSR codes
  - Local Parity: MSR codes → Fast Recovery and high performance on decoding process
  - Global parity: RS code → low cost and high fault tolerance
- AZ-Code: AZ(k,z,p,g)
  - k: the number of data blocks
  - z: the number of AZs
  - p: the number of local parity blocks
  - g: the number of global parity blocks



A 3AZ Example of AZ-Code

- AZ-Code:
  - k = 6, z=3, p=2, g = 3
- Local Parities:
  - MSR(4,2,4)
- Global Parities:
  - RS(6,3)
- We use this configuration as an example in our paper.

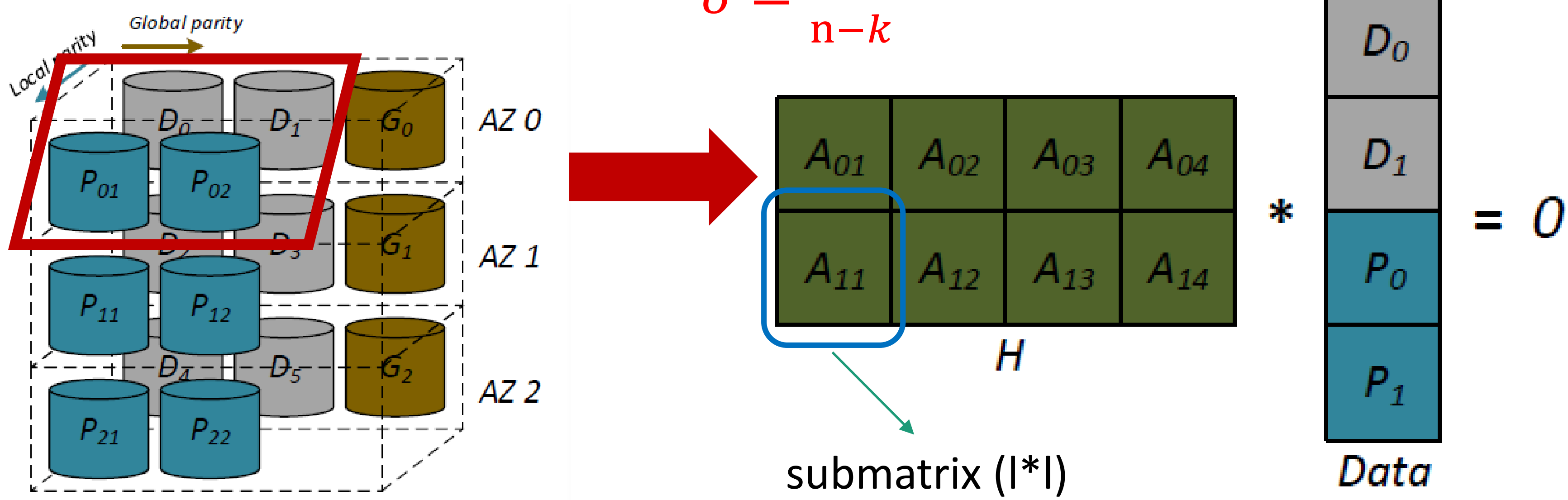


AZ(6,3,2,3) in 3AZ

Construction of AZ-Code: Local Parity

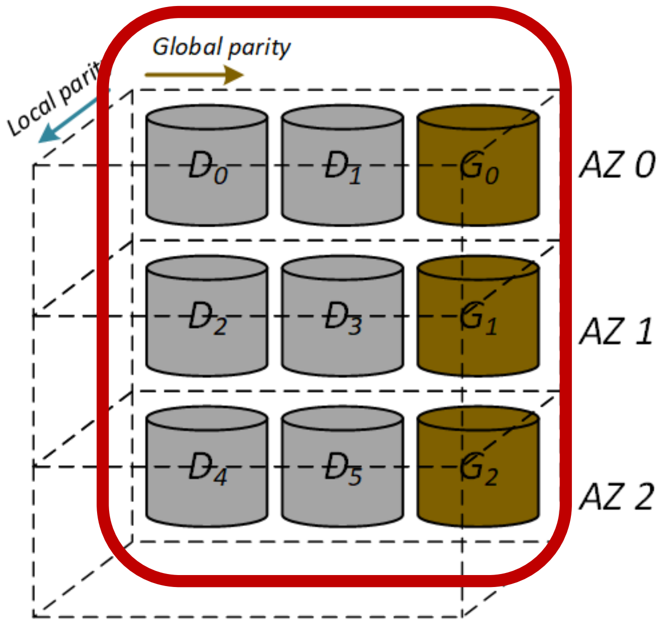
- Configuration: n = 4,k = 2,r = 2,s = m = 2,l = 4.
  - H: encoding matrix with the size of (r*l) * (n*l).
    Any r*r submatrix reversible
  - A_{ij}: l*l submatrix.
- Single node recovery cost of MSR：

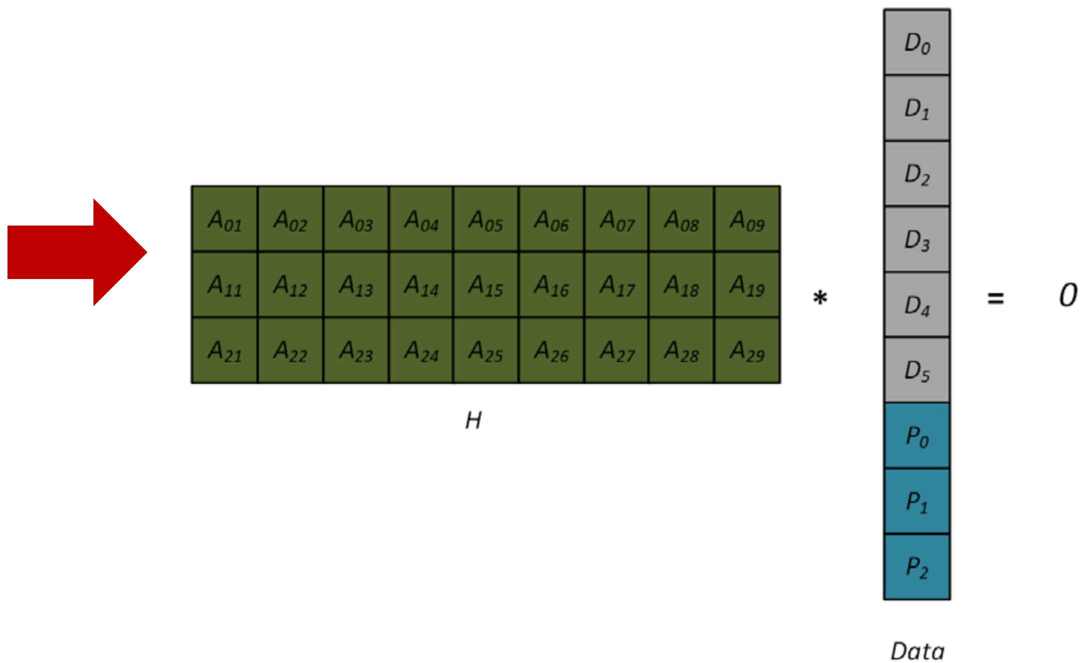
$$\delta = \frac{n}{n-k}$$



Construction of AZ-Code: Global Parity

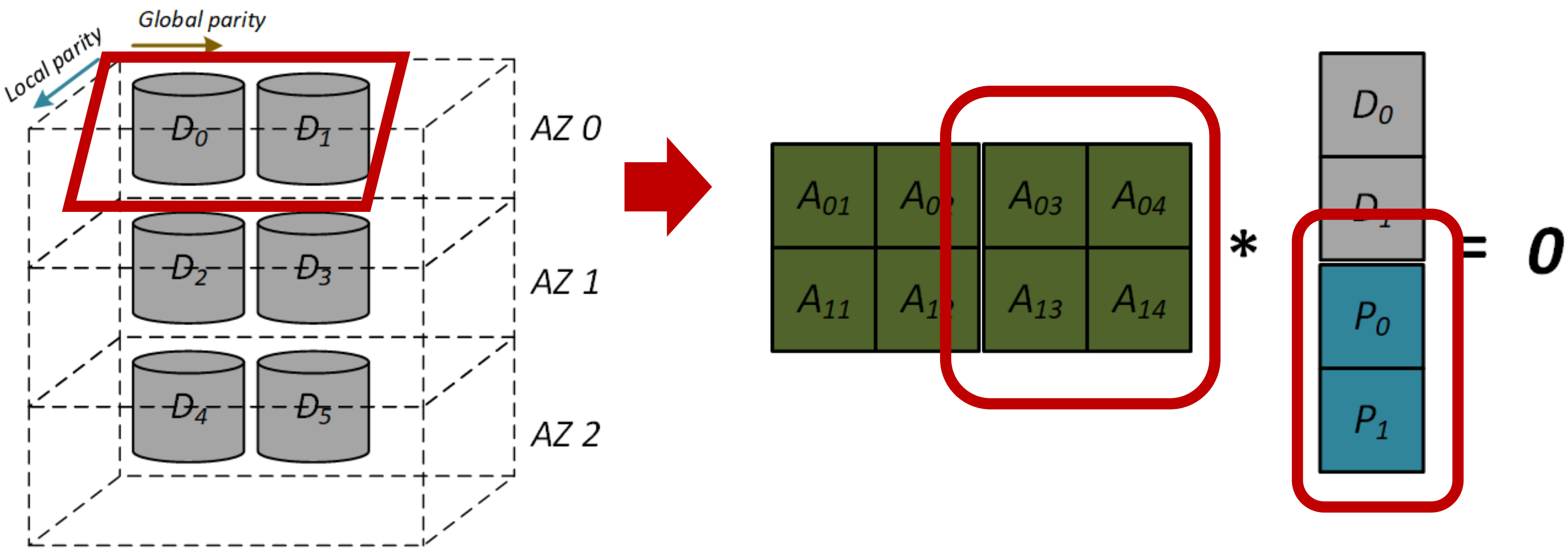
- RS Code (PCM method): n = 9,k = 6,r = 3
  - H: encoding matrix with the size of r * n. Any r*r submatrix reversible
  - A_{ij}: a number on Galois Field.





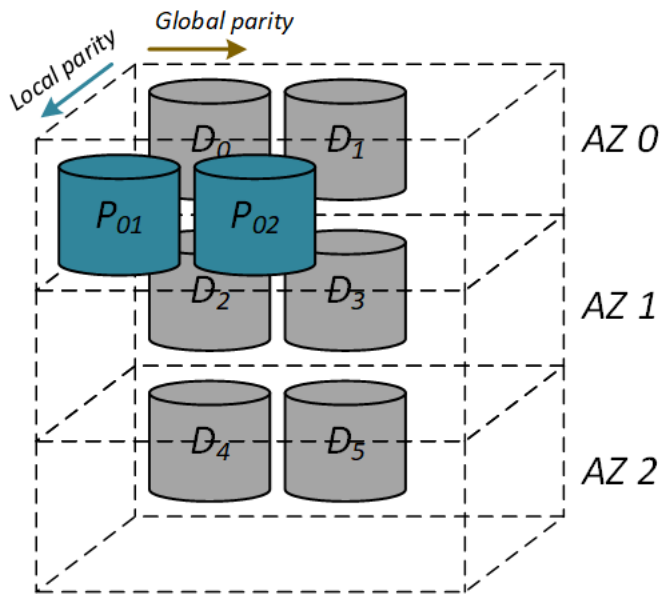
Construction of AZ-Code: PCM[SRDS'15](1)

- Encoding Local Parities: Parity Check Matrix(PCM)
  - PCM method can significantly reduce the size of matrix on encoding and decoding process.



Construction of AZ-Code: PCM[SRDS'15](1)

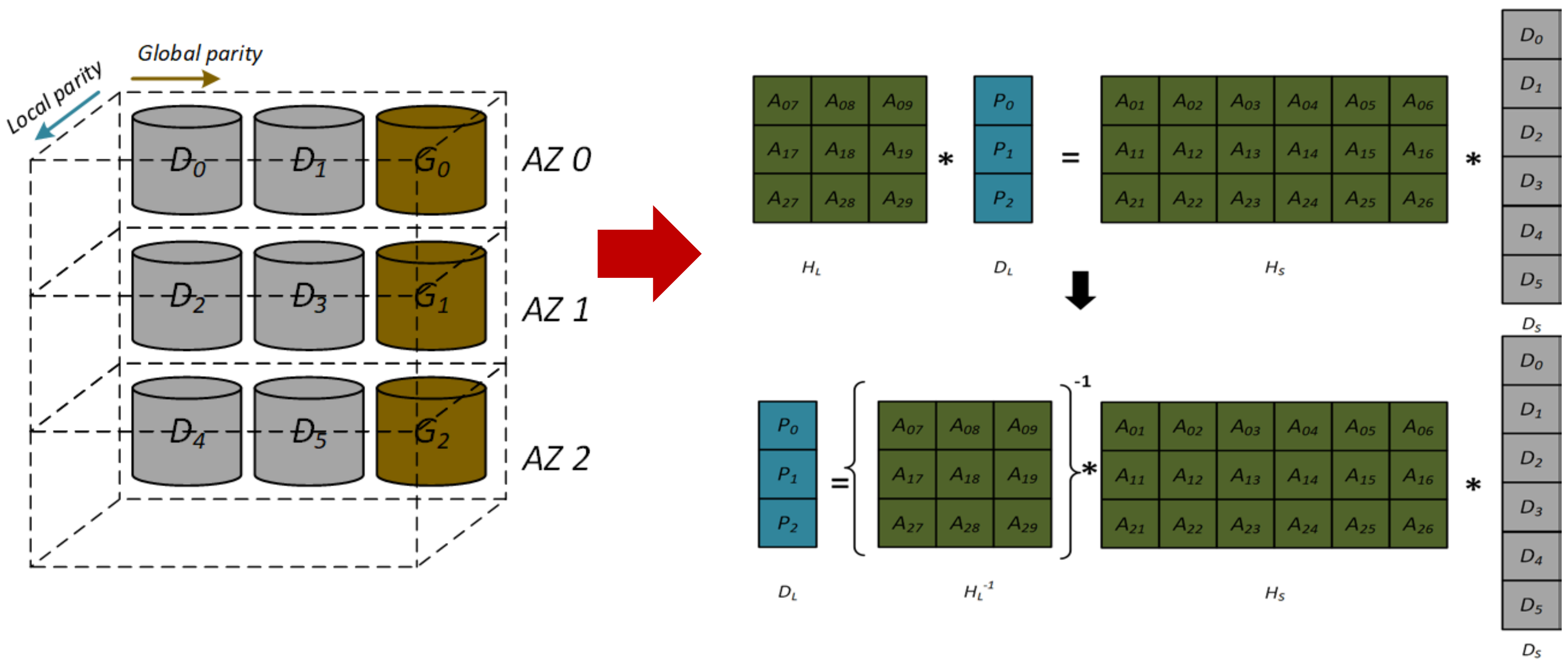
- Encoding Local Parities: Parity Check Matrix(PCM)
  - PCM method can significantly reduce the size of matrix on encoding and decoding process.



$$\underbrace{\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}}_{D_L} = \left\{ \underbrace{\begin{bmatrix} A_{03} & A_{04} \\ A_{13} & A_{14} \end{bmatrix}}_{H_L^{-1}} \right\}^{-1} * \underbrace{\begin{bmatrix} A_{01} & A_{02} \\ A_{11} & A_{12} \end{bmatrix}}_{H_S} * \underbrace{\begin{bmatrix} D_0 \\ D_1 \end{bmatrix}}_{D_S}$$

Construction of AZ-Code: PCM(2)

- Encoding Global Parities: Parity Check Matrix(PCM)



AZ-Code Reconstruction: three conditions

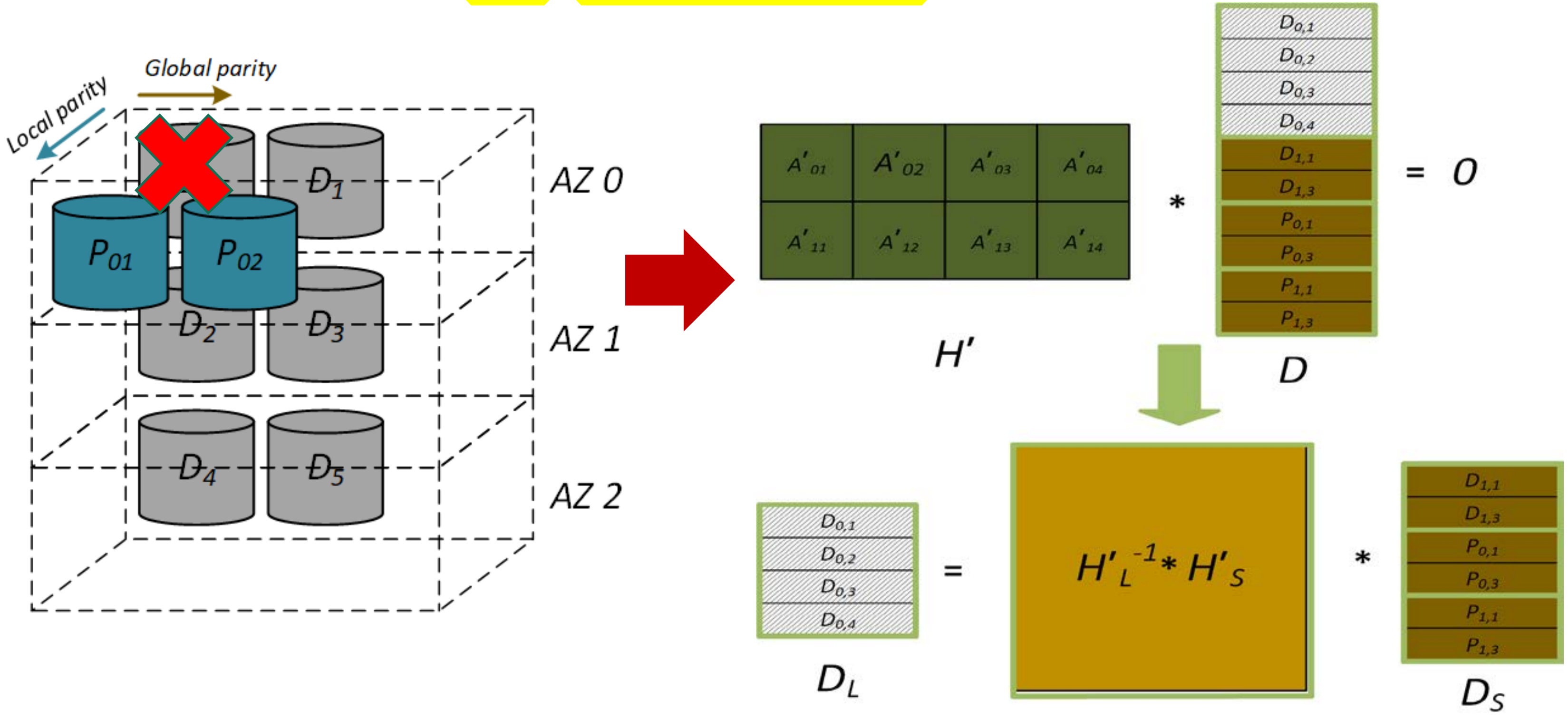
1. Decoding with Local Parities:
   - Single node failure: reduce the recovery cost
   - Multiple node failure: similar to RS code

2. Decoding with Global Parities

3. Decoding with Global and Local Parities (Combined method).

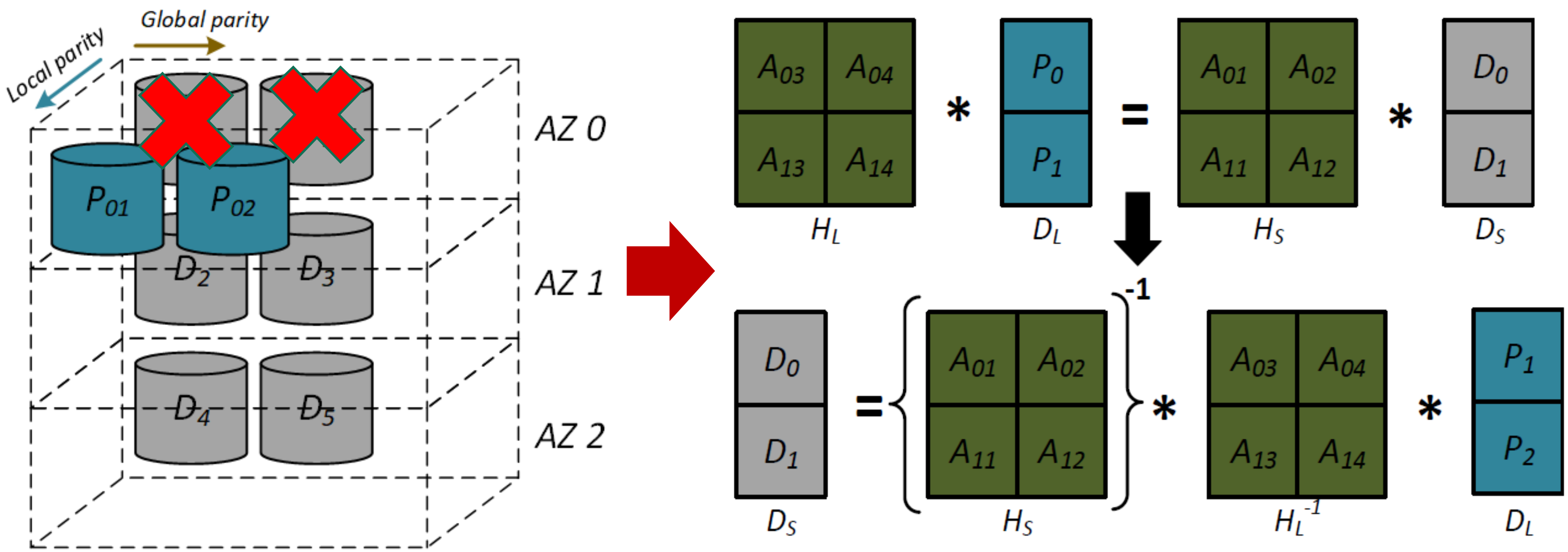
Reconstruction with Local Parities (1)

- Single node failure: Assume D_0 failed, the other three nodes D_1, P_{01} and P_{02} are used to recover the lost data together, and each node only need 1/2 data (marked as brown).



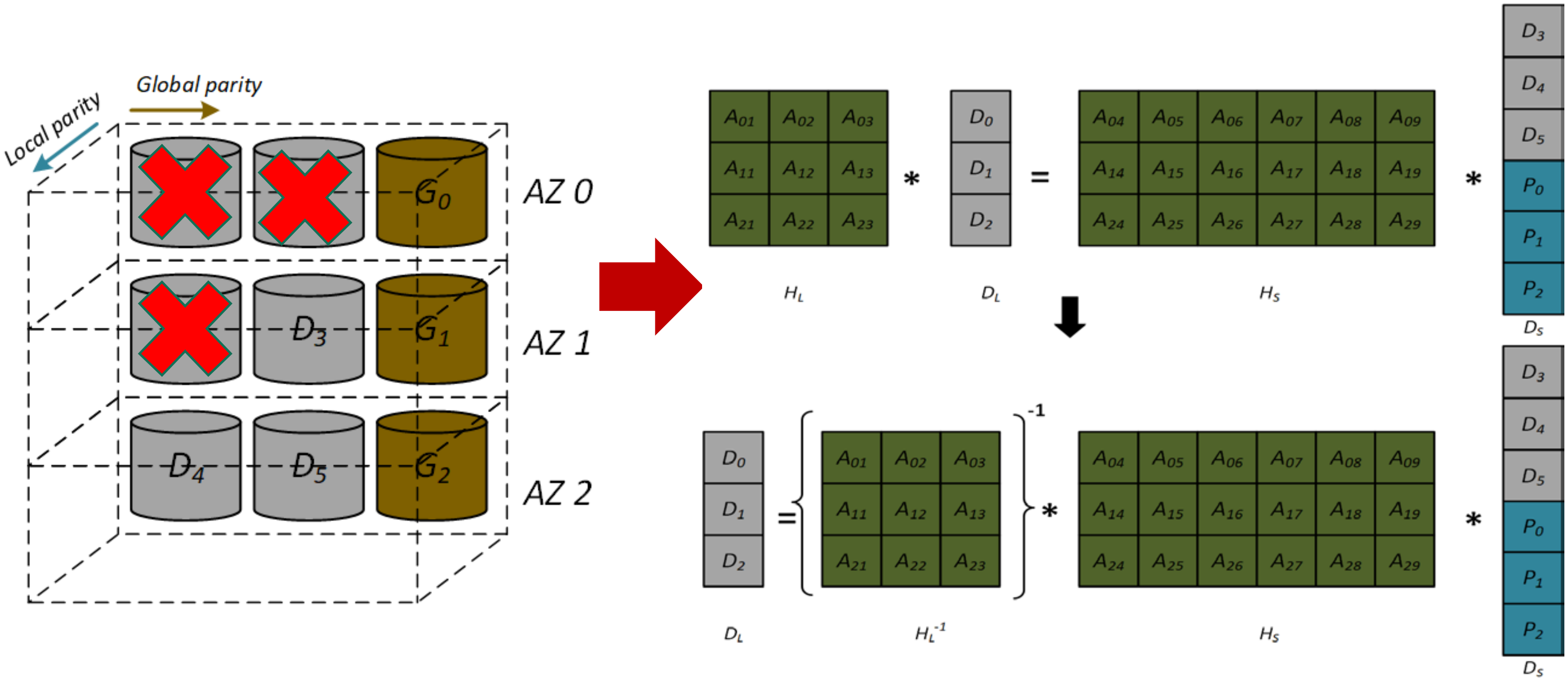
Reconstruction with Local Parities (2)

- Multiple node failures: Assume D_0 and D_1 failed. Use P_{01} and P_{02} to recover. Each node need all data



Reconstruction with Global Parities

- Arbitrary node-failures: Assume D_0 , D_1 and D_2 failed. Use Parity Check Matrix (PCM) to recover.



Reconstruction by Local and Global Parities concurrently(1)

- Decoding with both local and Global parities: Parity Check Matrix(PCM)
- J: new encoding matrix
- G is global parities and P is local parities

$$\underbrace{\begin{bmatrix} R_{01} & R_{02} & R_{03} & R_{04} & R_{05} & R_{06} & 1 & 0 & 0 & 0 & 0 \\ R_{11} & R_{12} & R_{13} & R_{14} & R_{15} & R_{16} & 0 & 1 & 0 & 0 & 0 \\ R_{21} & R_{22} & R_{23} & R_{24} & R_{25} & R_{26} & 0 & 0 & 1 & 0 & 0 \\ A_{01} & A_{02} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{03} & A_{04} \\ A_{11} & A_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{13} & A_{14} \end{bmatrix}}_{J} * \underbrace{\begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ G_0 \\ G_1 \\ G_2 \\ P_0 \\ P_1 \end{bmatrix}}_{Data} = 0$$

Reconstruction by Local and Global Parities concurrently(2)

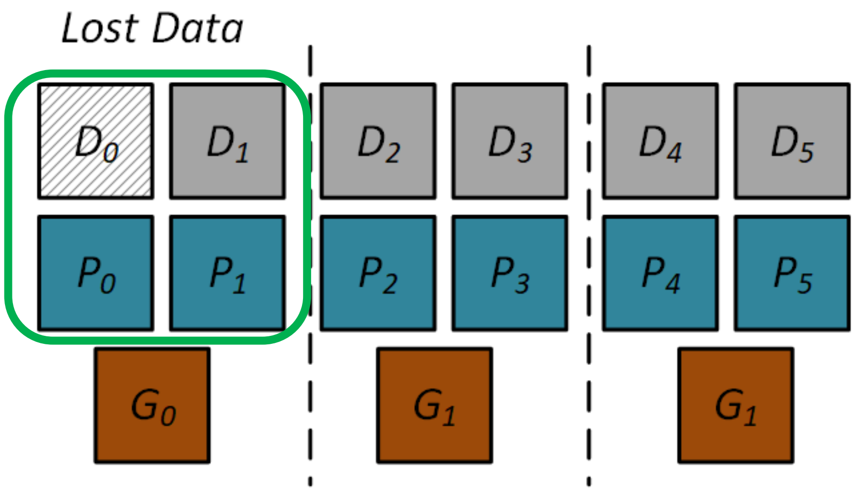
- Assume that D_0 , D_1, P_0 , P_1 and G_1 failed. The process is the same as before.

$$\underbrace{\begin{bmatrix} R_{01} & R_{02} & 1 & 0 & 0 \\ R_{11} & R_{12} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 & 0 \\ A_{01} & A_{02} & 0 & A_{03} & A_{04} \\ A_{11} & A_{12} & 0 & A_{13} & A_{14} \end{bmatrix}}_{H_L} * \underbrace{\begin{bmatrix} D_0 \\ D_1 \\ G_0 \\ P_0 \\ P_1 \end{bmatrix}}_{D_L} = \underbrace{\begin{bmatrix} R_{03} & R_{04} & R_{05} & R_{06} & 0 & 0 \\ R_{13} & R_{14} & R_{15} & R_{16} & 1 & 0 \\ R_{23} & R_{24} & R_{25} & R_{26} & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{H_S} * \underbrace{\begin{bmatrix} D_2 \\ D_3 \\ D_4 \\ D_5 \\ G_1 \\ G_2 \end{bmatrix}}_{D_S}$$

Reconstruction Cost Analysis: decoding by local parities (1)

- Single Node Fault: using Local Parity for recovery. recovery cost:3/2=1.5

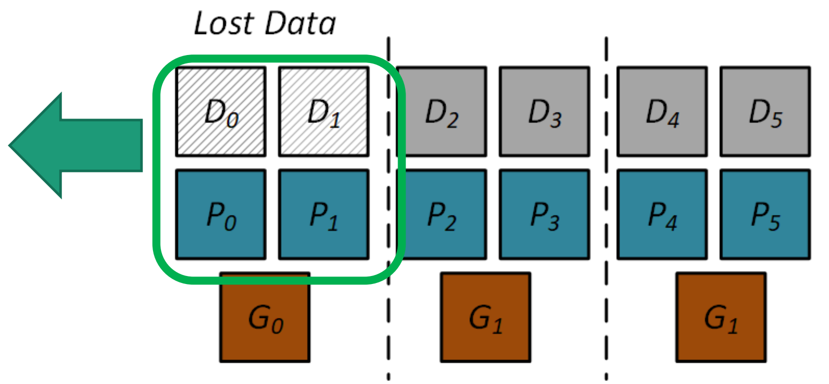
- Using the 3 survivors for recovery
- Each node only needs to transmit half of its data



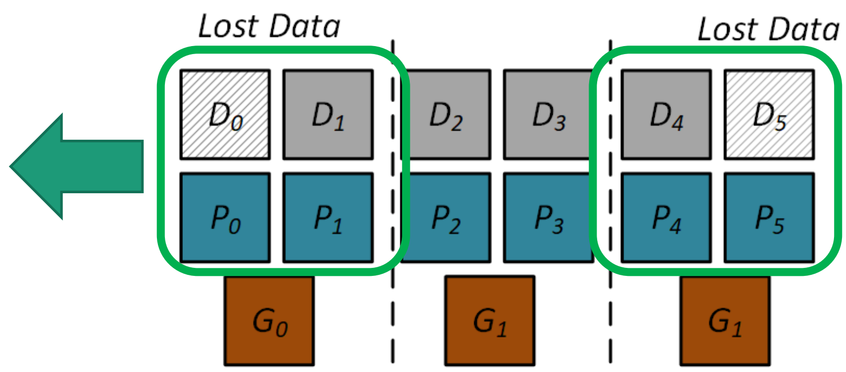
Reconstruction Cost Analysis: decoding by local parities (2)

- Two Node Fault: using Local Parity for recovery.

Recovery cost:2



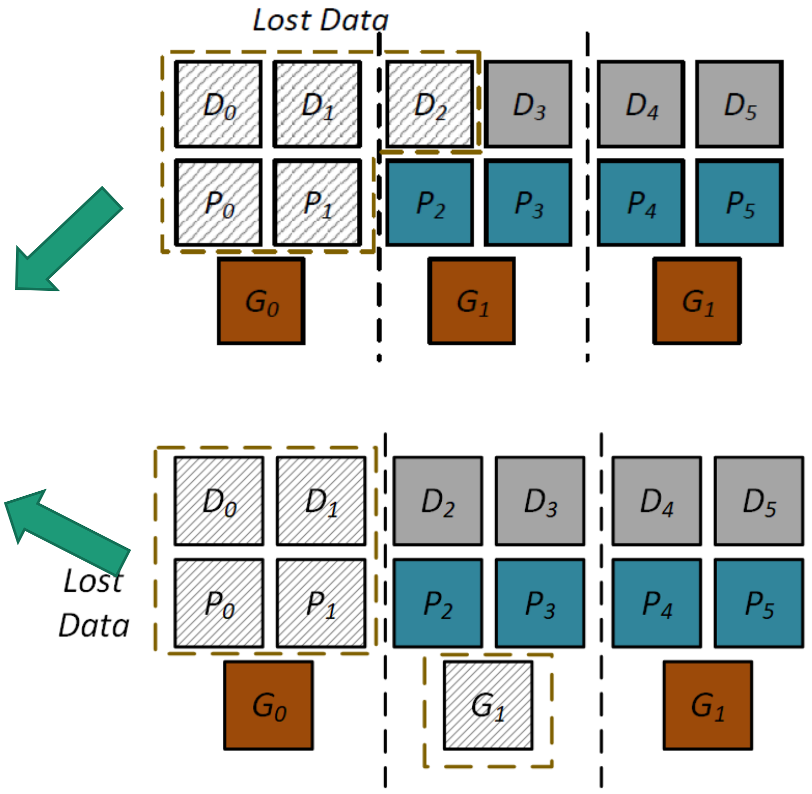
Recovery cost:3



Reconstruction Cost Analysis: decoding by Global parities

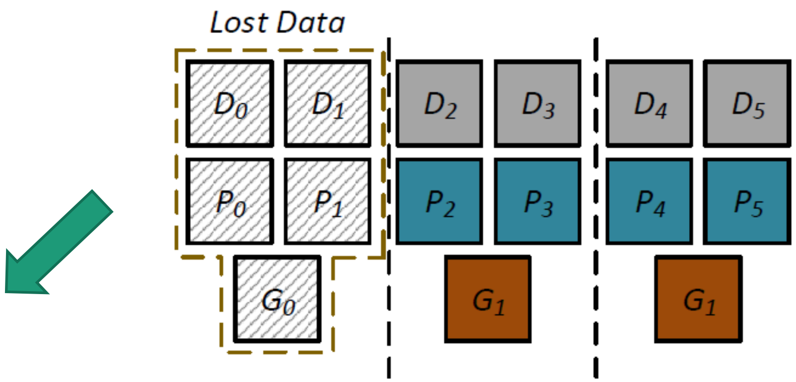
- Multiple Node Fault : using global parity

Recovery cost:6

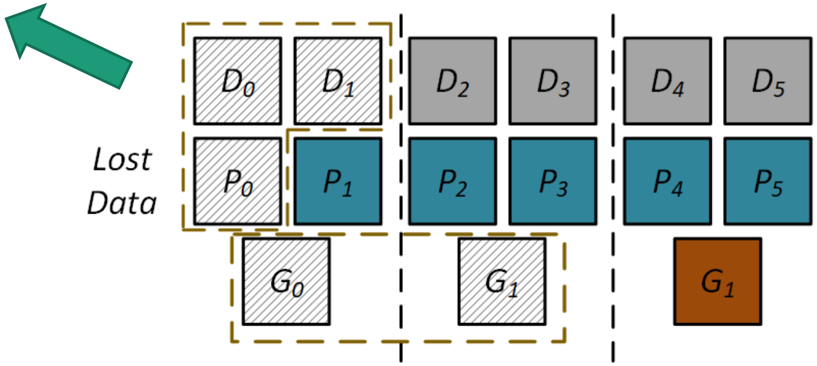


Reconstruction Cost Analysis: decoding by local and global parities

- Multiple Node Fault : using both local and global parity for recovery.



Recovery cost:6



Configuration of AZ-Code

- By changing the erasure code parameters configuration, we can trade off storage cost, fault tolerance and single-node recovery bandwidth.

	Storage Efficiency	Fault Tolerance (maximum concurrent fail nodes)	Recovery Cost (single node)
AZ(k ,z ,p , g)	k/(k+z*p+g)	g + p	(k/z+p-1)/p
AZ(6,3,2,3)	0.40	1AZ (5 nodes)	1.5
AZ(12,3,2,6)	0.50	1AZ (8 nodes)	2.5
AZ(12,3,2,9)	0.44	1AZ+2 (11 nodes)	2.5
AZ(18,3,2,6)	0.60	1AZ-2 (8 nodes)	3.5
AZ(18,3,2,9)	0.55	1AZ(11 nodes)	3.5
AZ(18,3,2,12)	0.50	1AZ+2 (14 nodes)	3.5

Lowest Recovery Cost

Best Storage Efficiency

Content

01 Background

02 AZ-Code

03 Evaluation

04 Conclusion

Evaluation Methodology

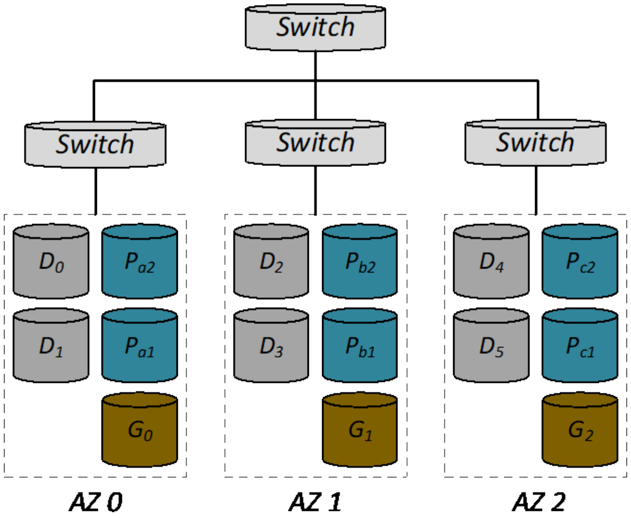
- Erasure codes in our evaluation
  - AZ-Code, EVENODD [ToC'95], STAR [FAST'05], RS code, LRC[ATC'2012], MSR Code[FAST'2018])
- Metric:
  - Reconstruction Cost: The number of nodes required to reconstruct the lost nodes
  - Recovery Time, including
    - computation time
    - I/O overhead
    - transmission time

Experimental Environment

- Build a Hadoop Cluster of three Data Nodes and one Name Node
- All Four nodes are DELL R730 Servers

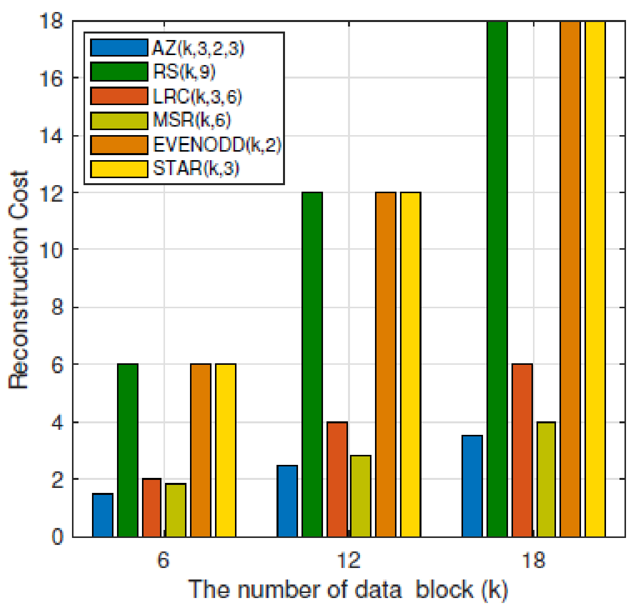
	Configuration
Operating System	Linux 3.19
C Complier	gcc 5.4.0
CPU	Intel Xeon 3.0GHz
Memory	32GB

DELL R730 Server

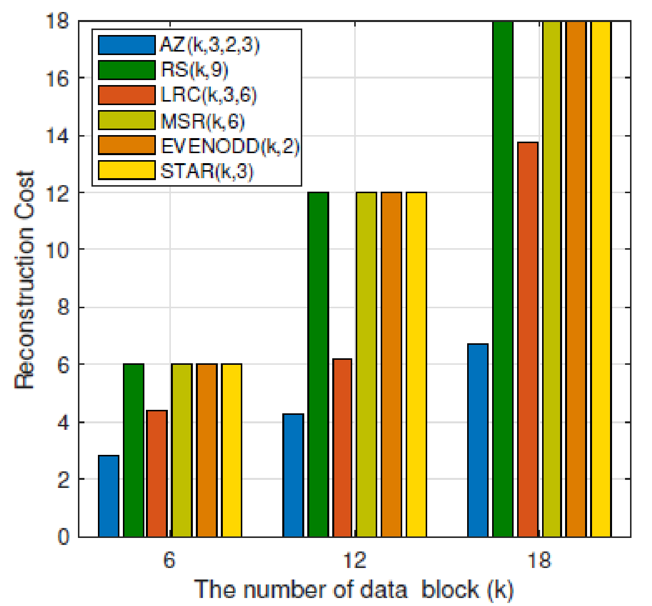


Mathematical analysis: reconstruction cost

- We Change k from 6 to 18 and consider single and double data node failure conditions.
  - save the reconstruction cost by up to 81% compared with RS code, EVENODD and STAR, 51% compared with LRC and 63% compared with MSR.



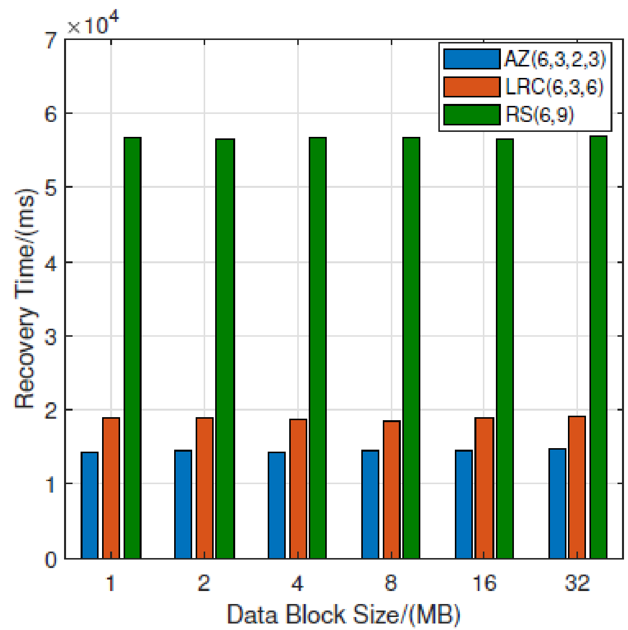
(a) Single Data Node Failure.



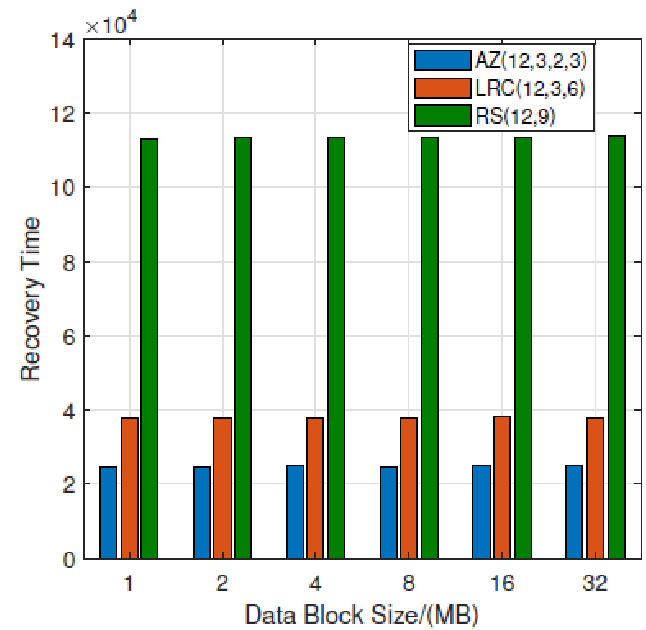
(b) Double Data Nodes Failure.

Experimental Results: Recovery time on Single Data Node Failure(1)

- Change k from 6 to 12 and block size from 1MB to 32 MB
  - save the recovery time by up to 34% compared with LRC
  - save the recovery time by up to 78% compared with RS



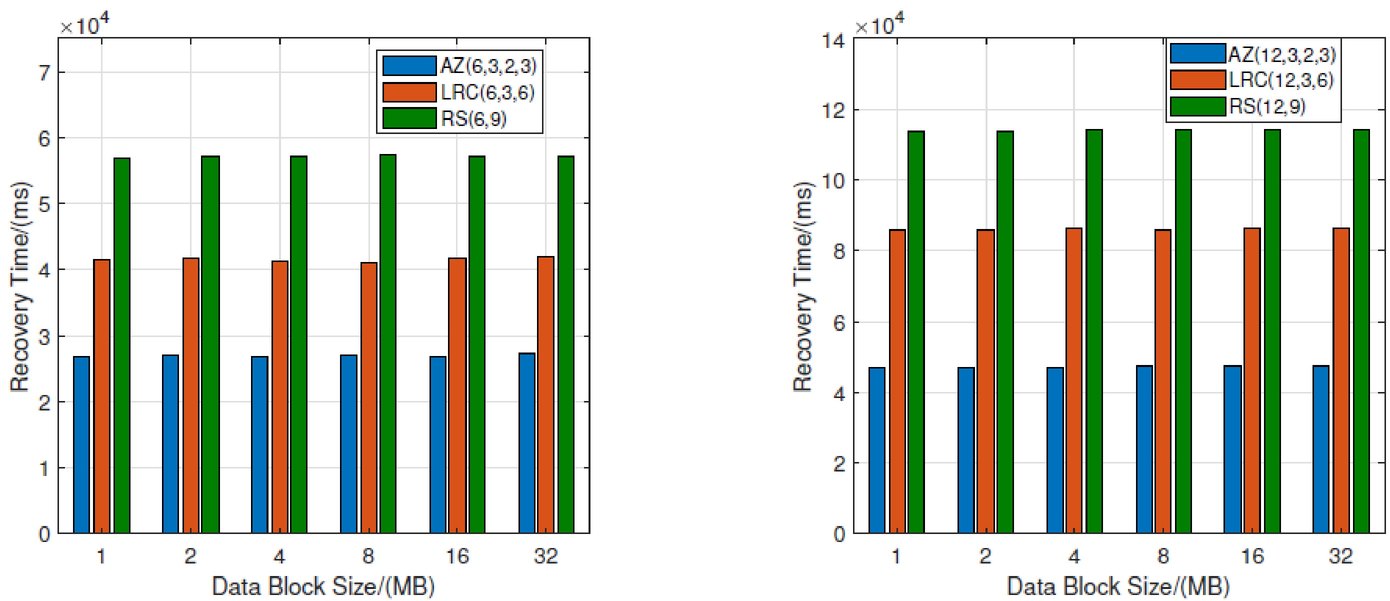
(a) k=6



(b) k=12

Experimental Results: Recovery time on Double Data Node Failure(2)

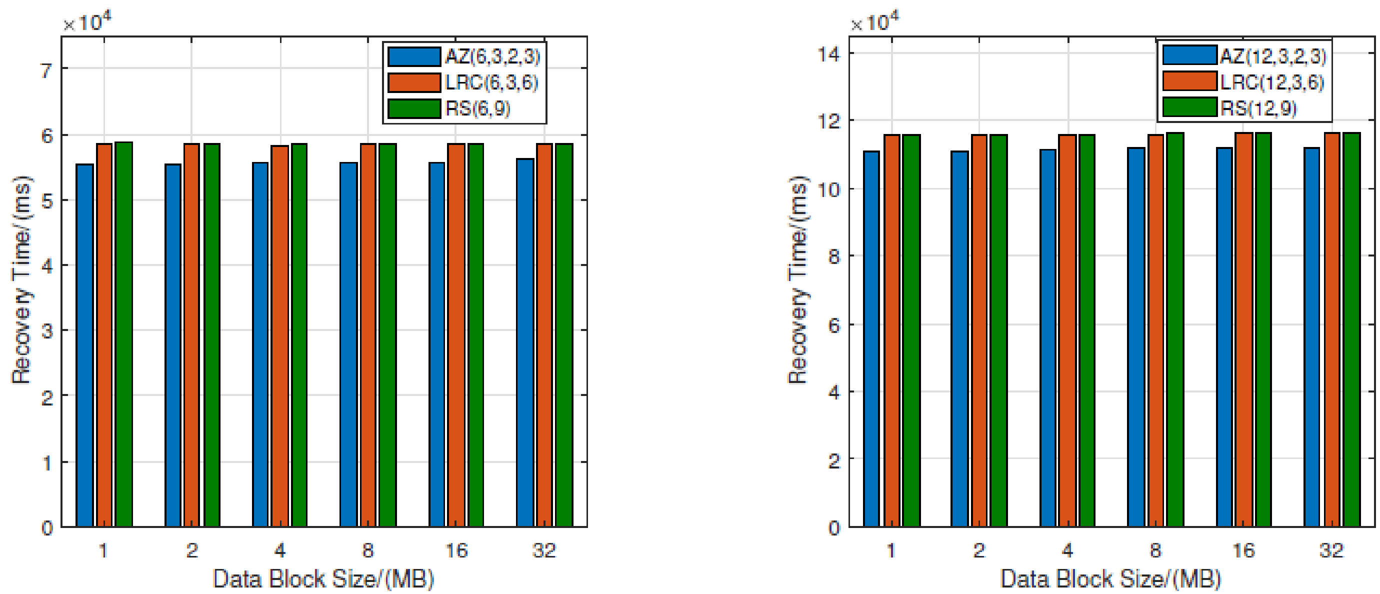
- Change k from 6 to 12 and block size from 1MB to 32 MB
  - save the recovery time by up to 45% compared with LRC
  - save the recovery time by up to 58% compared with RS



(a) k=6 (b) k=12

Experimental Results: Recovery time on Double Data Node Failure(2)

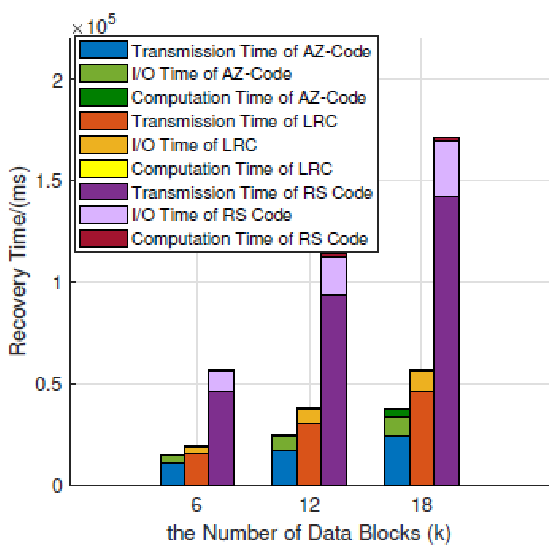
- Change k from 6 to 12 and block size from 1MB to 32 MB
    - save the recovery time by up to 5% compared with LRC
    - save the recovery time by up to 6% compared with RS



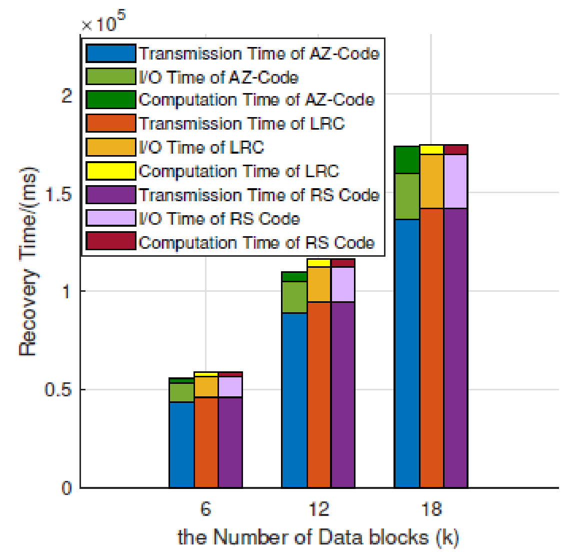
(a) k=6 (b) k=12

Experimental Results of Recovery Time

- We analysis the components of recovery time
  - transmission time is about 8.71x of computation time
  - transmission time is about 5.45x of I/O overhead
  - transmission time is the major component of recovery time



(a) Detailed recovery time of full system recovery in terms of Single node failure



(b) Detailed recovery time of full system recovery in terms of Multiple nodes failure

Content

01 Background

02 AZ-Code

03 Evaluation

04 Conclusion

Conclusion

- Availability Zone Level Erasure Code (AZ-Code) has the following advantages
  - Supporting the Multiple AZ environment well and provide AZ level reliability.
  - Improving the recovery performance in AZ environment
    - reducing the recovery time up to 78.24% compared with RS code
    - reducing the recovery time up to 34.81% compared with LRC code
  - Ensuring high scalability

Thank you!

Q & A