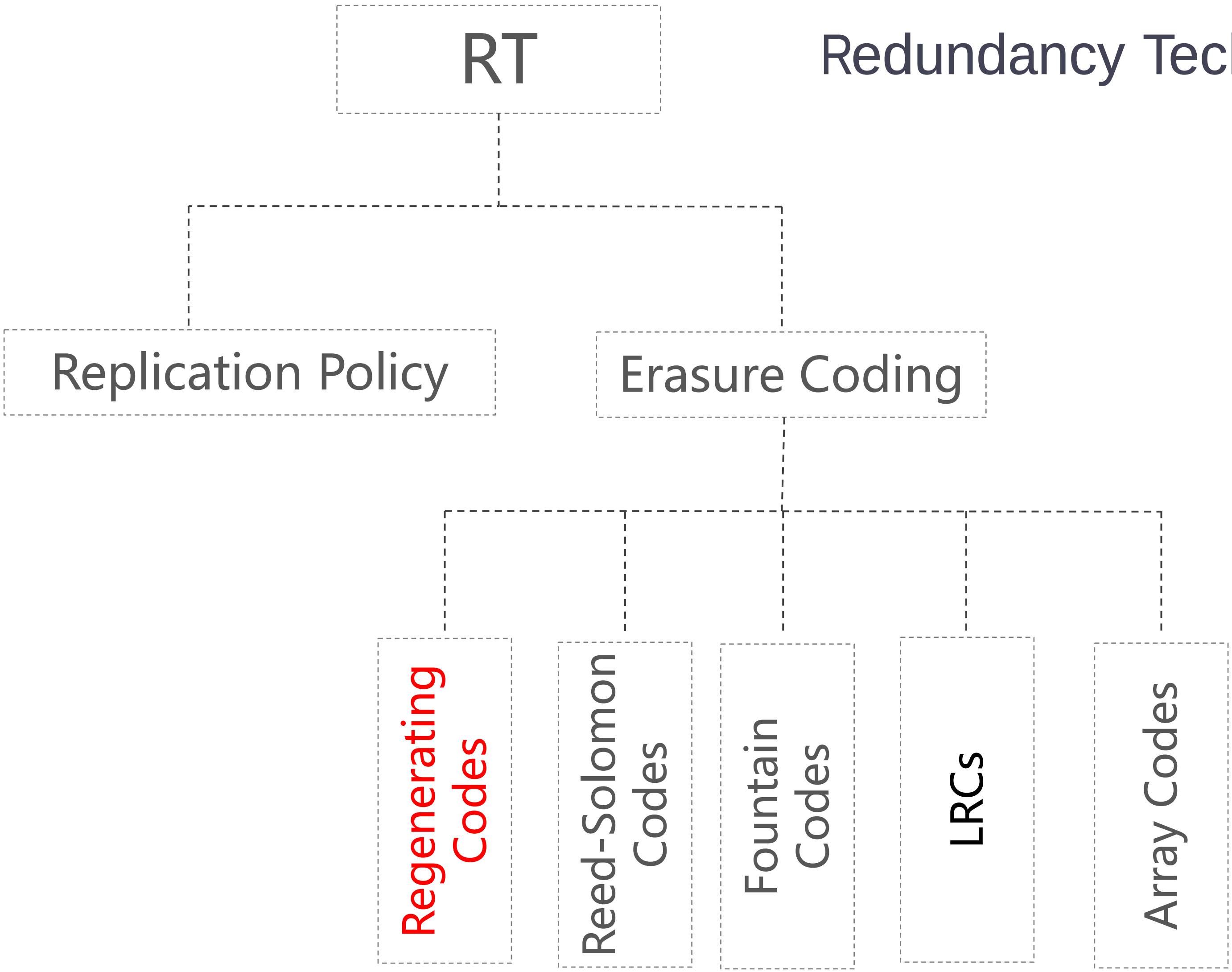
Linear Exact-Repair Construction of Hybrid MSR Codes in Distributed Storage System

Haibin Kan（阚海斌）
School of Computer Science , Fudan University, China

Redundancy Technique in DSS



Storage Policies

A Scenario using Erasure Codes for DSS

There are three kinds of Video data： 1） Metadata； 2） HD Video data； 3） Others. The metadata are with huge volume, need high availability， but lower access speed and concurrent access number. The others are obtained by transformation of HD Video data. Based on the above requirement, the storage policies are as follows：

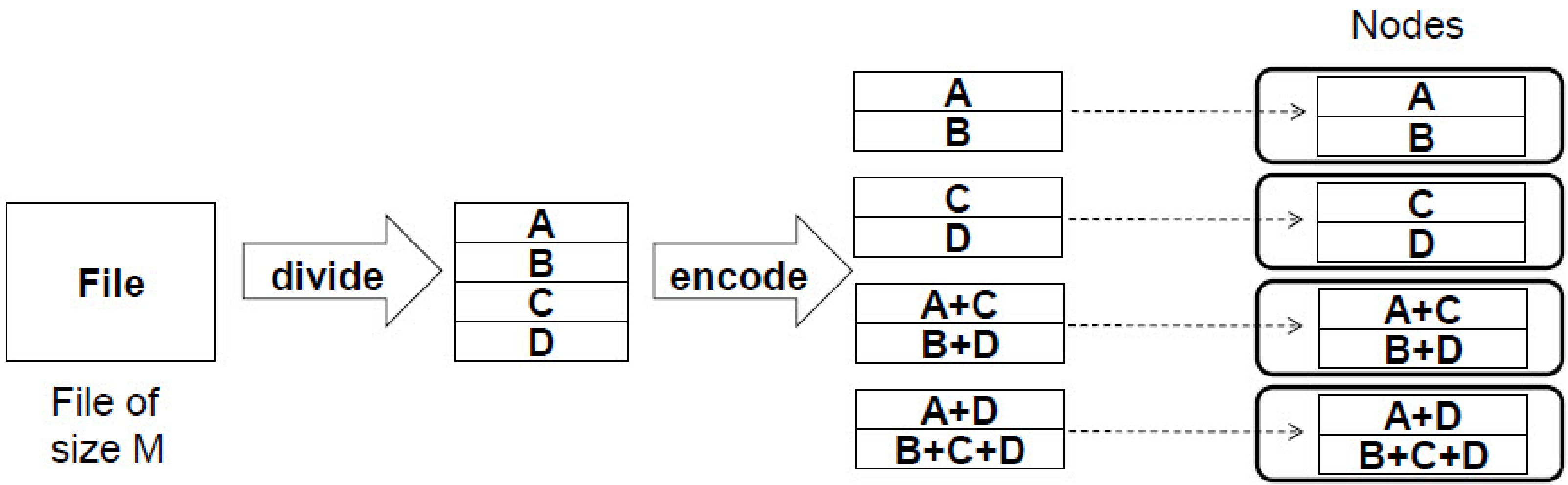
① Use RS code, *RS*(7, 4) code, for Metadata. Divided one metadat file into 4 bulks， and generate 3 parity bulks. The storage efficiency is 57%；

② Use 3x storage policy for HD Video data, and the storage efficiency is 33%.

③ Use 2x storage policy for the others, and the storage efficiency is 50%.

Traditional Repair

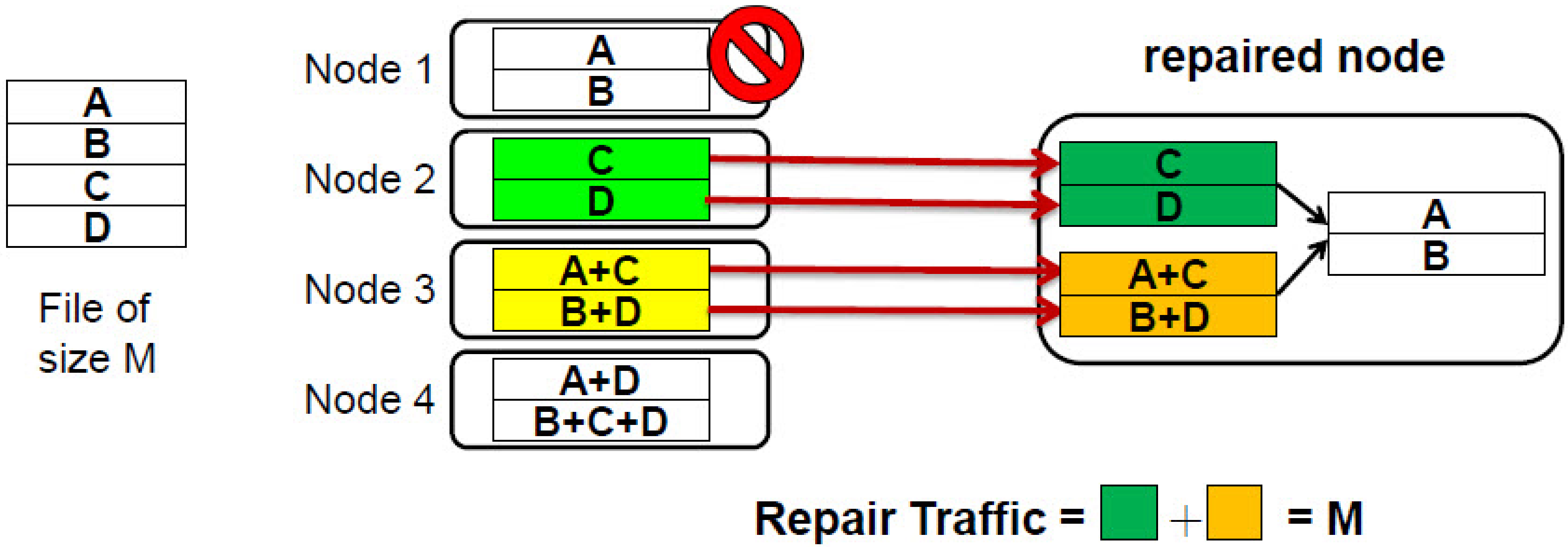
Example: a (4,2) MDS distributed storage code



n = 4, k = 2

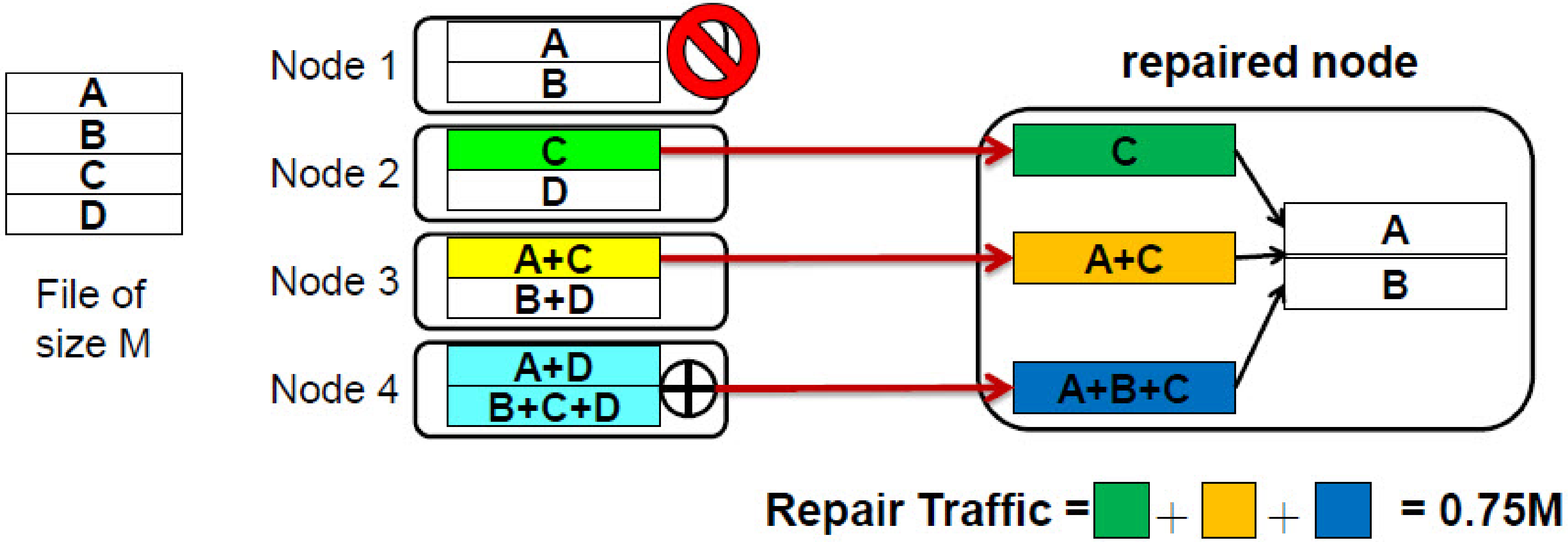
Traditional Repair

- **Conventional repair**: download data from any k nodes



Repair by Regerating Codes

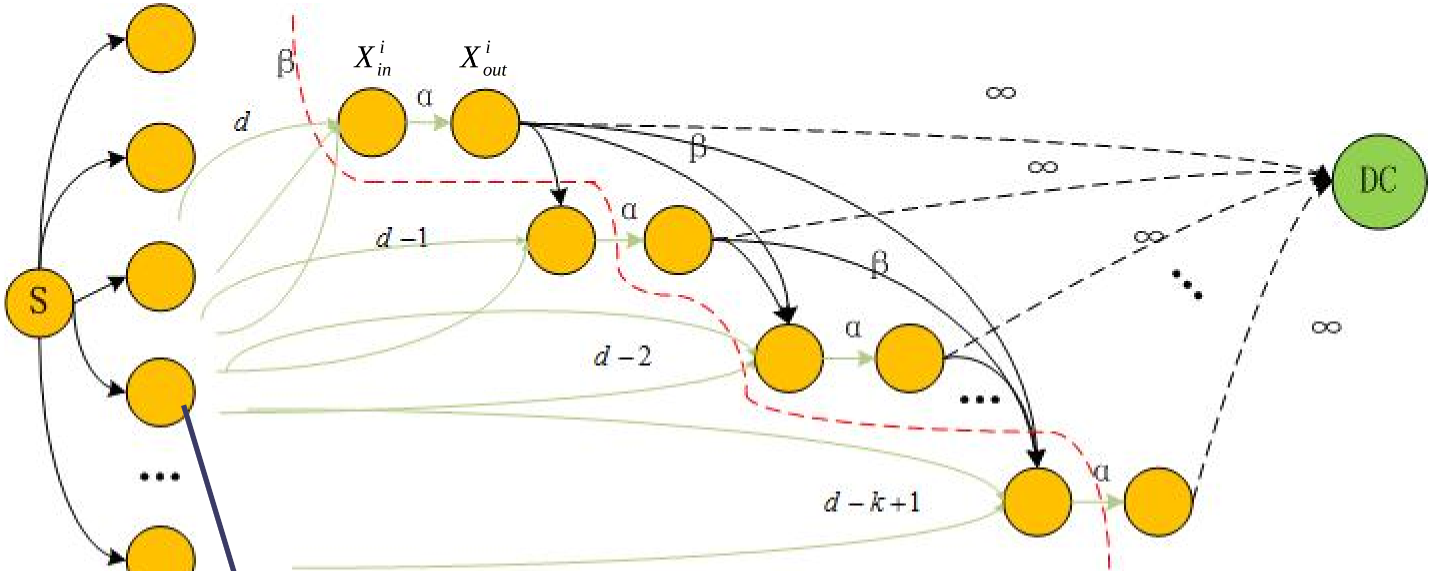
- Repair in regenerating codes:
  - Surviving nodes encode chunks (network coding)
  - Download one encoded chunk from each node



- *Minimizing repair traffic → minimizing system downtime*

Distributed Storage Systems

Repair Procedure Analysis



Information flow graph $G(n, k, d; \alpha, \beta, B)$. Let $\gamma = d\beta$

$$\boxed{B \leq \sum_{i=0}^{k-1} \min\{(d-i)\beta, \alpha\}}$$

[1] Dimakis A G, Godfrey P B, Wu Y, et al. Network coding for distributed storage systems[J]. Information Theory, IEEE Transactions on, 2010,56(9): 4539-4551

The tradeoff between storage size α and repair bandwidth γ.

MSR(Minimum Storage Regenerating)：

$$(\alpha,\gamma)=\left(\frac{B}{k},\frac{Bd}{k(d-k+1)}\right)$$

MBR(Minimum Bandwidth Regenerating)：

$$(\alpha,\gamma)=\left(\frac{2Bd}{2kd-k^2+k},\frac{2Bd}{2kd-k^2+k}\right)$$

[1] → [2] → [3] → Problem

When $\beta=1$，$d<2k-3$ There exist no MSR code by exact repair model.

Interference Alignment

When $d \geq 2k-1$, MISER Code.

Product Matrix

When $d \geq 2k-2$, The Construction of MSR. Arbitrary (n, k, d) MBR Code.

1. High Complexity for decoding and repairing.
2. High repair degree
3. High repair delay

[1] Dimakis A G, Godfrey P B, Wu Y, et al. Network coding for distributed storage systems[J]. Information Theory, IEEE Transactions on, 2010,56(9): 4539-4551

[2] Shah N B, Rashmi K V, Kumar P V, et al. Interference alignment in regenerating codes for distributed storage: Necessity and code constructions[J]. Information Theory, IEEE Transactions on, 2012, 58(4): 2134-2158

[3] Rashmi K V, Shah N B, Kumar P V. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction[J]. Information Theory, IEEE Transactions on, 2011, 57(8):5227-5239

Without loss of generality, let $\beta = 1$

$$MSR: \begin{cases} \alpha = d - k + 1 \\ B = k(d - k + 1) \end{cases}$$

$$MBR: \begin{cases} \alpha = d \\ B = kd - \binom{k}{2} \end{cases}$$

Product Matrix

$$\underbrace{C}_{n\times\alpha} = \underbrace{\Psi}_{n\times d} \times \underbrace{M}_{d\times\alpha}$$

- $\boldsymbol{C}$: Code Matrix
  - Every row represent one node
  - α symbols stored in i^{th} node
- Ψ : Coding Matrix
  - Prepared before encoding
- M : Message Matrix
  - Contains the B source symbols, with some repeated symbols

Product Matrix

- Construction of [n,k,d] MBR code, $\alpha = d,\ B = \binom{k+1}{2} + k(d-k),\ \beta = 1$
- Explicit MSR Code for all n, k, d

$$\underbrace{C}_{n\times d} = \underbrace{\Psi}_{n\times d} \times \underbrace{M}_{d\times d}$$

- Message Matrix(Symmetric)

$$M = \begin{bmatrix} \underbrace{S}_{k\times k} & \underbrace{T}_{k\times(d-k)} \\ \underbrace{T^t}_{(d-k)\times k} & \underbrace{0}_{(d-k)\times(d-k)} \end{bmatrix}, \quad \begin{bmatrix} u_1 \\ \vdots \\ u_{k(k+1)/2} \\ u_{k(k+1)/2+1} \\ \vdots \\ u_B \end{bmatrix}$$

- Encoding Matrix

$$\underbrace{\Psi}_{n\times d} = [\underbrace{\Phi}_{n\times k} \quad \underbrace{\Delta}_{n\times(d-k)}]$$

Φ : any k rows linearly independent
Ψ : any d rows linearly independent

Product Matrix

When node f is failed, we need to recover $\varphi_f^t M$

Helper node i passes: $\varphi_i^t M \varphi_f$

After receive data from d nodes: $\Psi_{(d \times d)} M \varphi_f$

$$\Psi_{(d \times d)} M \varphi_f \xrightarrow{\Psi_{(d\times d)} \text{ is invertible}} M \varphi_f \xrightarrow{M \text{ is symmetric}} \varphi_f^t M$$

Similarly, we can do data-reconstruction at each DC

Product Matrix

- Construction of [n,k,d=2k-2] MSR code, $\alpha = k-1 = d/2, B = \alpha(\alpha+1), \beta = 1$

$$\underbrace{C}_{n\times\alpha} = \underbrace{\Psi}_{n\times d} \times \underbrace{M}_{d\times\alpha}$$

- Message Matrix(S_1, S_2 are symmetric)

$$M = \begin{bmatrix} \underbrace{S_1}_{\alpha\times\alpha} \\ \underbrace{S_2}_{\alpha\times\alpha} \end{bmatrix}, \quad \begin{bmatrix} u_1 \\ \vdots \\ u_{\alpha(\alpha+1)/2} \\ u_{\alpha(\alpha+1)/2+1} \\ \vdots \\ u_B \end{bmatrix}$$

- Encoding Matrix

$$\underbrace{\Psi}_{n\times d} = [\underbrace{\Phi}_{n\times\alpha} \quad \underbrace{\Lambda\Phi}_{n\times\alpha}] \quad \Lambda : n\times n \ diagonal \ matrix$$

Φ : any α rows linearly independent
Ψ : any d rows linearly independent
Λ : the diagonal elements are distinct

Product Matrix

When node *f* is failed, we need recover $[\phi_f^t \ \ \lambda_f \phi_f^t]M = \phi_f^t S_1 + \lambda_f \phi_f^t S_2$

Helper node *i* passes: $\varphi_i^t M \phi_f$

After receive data from *d* nodes: $\Psi_{(d\times d)} M \phi_f$

$$\Big\downarrow \ \Psi_{(d\times d)} \text{ is invertible}$$

$$M\phi_f = \begin{bmatrix} S_1\phi_f \\ S_2\phi_f \end{bmatrix}$$

$$\Big\downarrow \ S_1, S_2 \text{ are symmetric}$$

$$\phi_f^t S_1 + \lambda_f \phi_f^t S_2$$

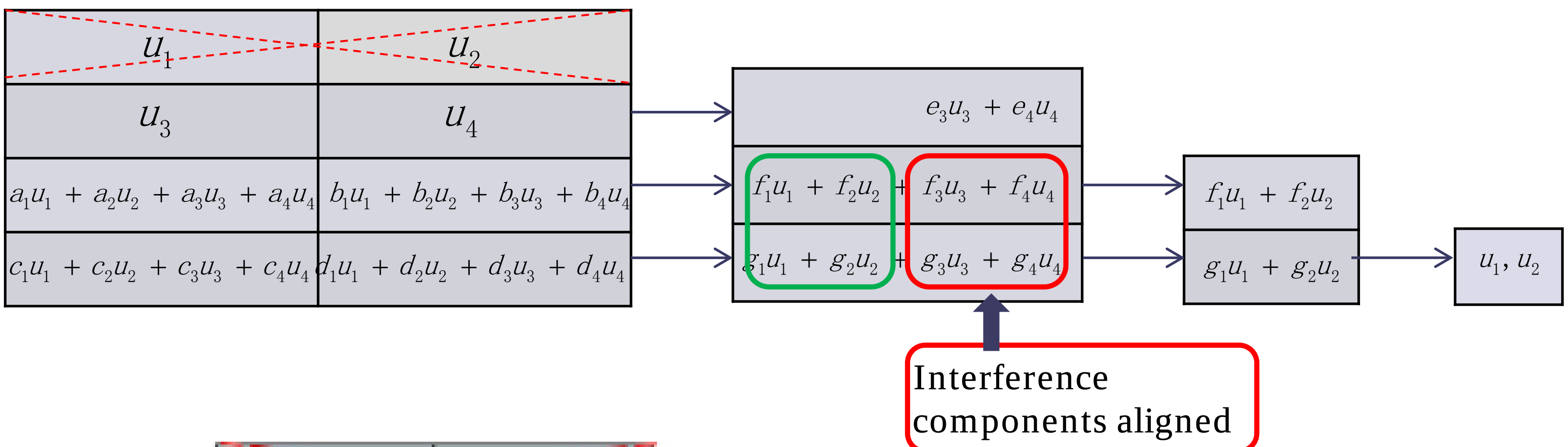
Similarly, we can make Data Reconstruction at each DC, where Λ is important.

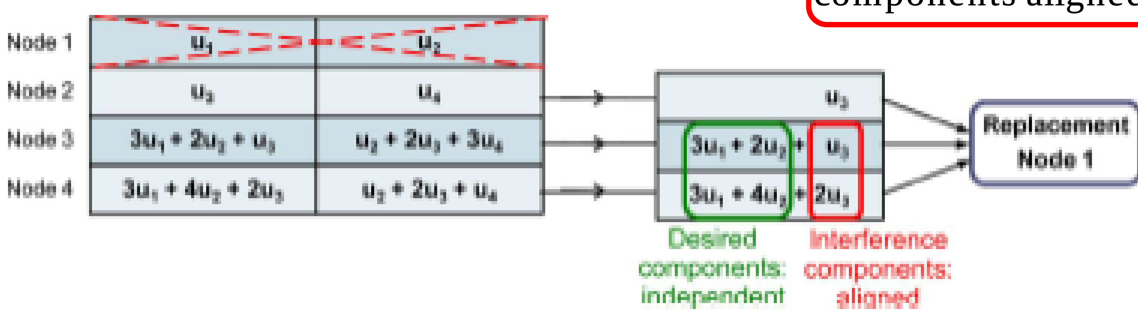
It is easy to construct [n,k,d>2k-2] MSR codes from [n,k,2k-2] MSR codes

Interference Alignment

MISER code: a systematic MDS code that achieves the lower bound on repair bandwidth for the exact repair of systematic nodes.

Systematic [4,2,3] MISER code, $\alpha = d - k + 1 = 2,\ B = k\alpha = 4, \beta = 1$





Interference Alignment

The construction of systematic [6,3,5] MISER code, $\alpha = d - k + 1 = 3,\ B = k\alpha = 9$

1) Encoding Matrix of each node

- Encoding matrix of node m $\quad \underbrace{G^m}_{B\times\alpha}$

- Systematic node $m \in \{1,2,3\}$ $\rightarrow$ $G^1 = \begin{bmatrix} I_{3\times3} \\ 0 \\ 0 \end{bmatrix} \quad G^2 = \begin{bmatrix} 0 \\ I_{3\times3} \\ 0 \end{bmatrix} \quad G^3 = \begin{bmatrix} 0 \\ 0 \\ I_{3\times3} \end{bmatrix}$

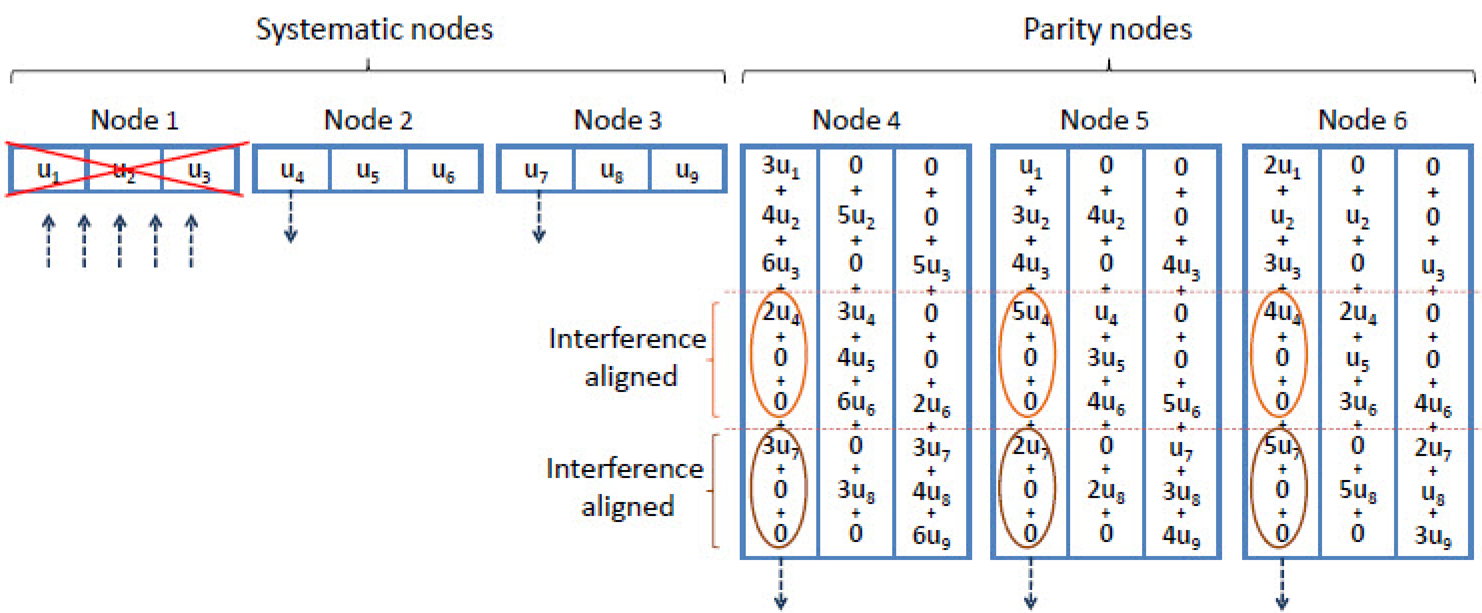
- Parity node $m \in \{4,5,6\}$

$$G^m = \left[\begin{array}{ccc} 2\varphi_1^{(m)} & 0 & 0 \\ 2\varphi_2^{(m)} & \varphi_1^{(m)} & 0 \\ 2\varphi_3^{(m)} & 0 & \varphi_1^{(m)} \\ \hline \varphi_2^{(m)} & 2\varphi_1^{(m)} & 0 \\ 0 & 2\varphi_2^{(m)} & 0 \\ 0 & 2\varphi_3^{(m)} & \varphi_2^{(m)} \\ \hline \varphi_3^{(m)} & 0 & 2\varphi_1^{(m)} \\ 0 & \varphi_3^{(m)} & 2\varphi_2^{(m)} \\ 0 & 0 & 2\varphi_3^{(m)} \end{array}\right], \quad where \begin{bmatrix} \varphi_1^4 & \varphi_1^5 & \varphi_1^6 \\ \varphi_2^4 & \varphi_2^5 & \varphi_2^6 \\ \varphi_3^4 & \varphi_3^5 & \varphi_3^6 \end{bmatrix} \text{ is Cauchy matrix}$$

Interference Alignment

The construction of systematic [6,3,5] MSR code, $\alpha = d - k + 1 = 3,\ B = k\alpha = 9$

For example, $Let\ \Psi = \begin{bmatrix} 5 & 4 & 1 \\ 2 & 5 & 4 \\ 3 & 2 & 5 \end{bmatrix}$, which is a Cauchy matrix *over* F_7.



2) Data Regeneration: To regenerate the i-th systematic node, each of the remaining nodes passes their respective i-th symbol.

Interference Alignment

- [2k,k,2k-1] MISER code, $\alpha = d - k + 1 = k,\ B = k\alpha = k^2$
- [n,k,n-1] MISER code, where n>=2k
- [n,k,d] MISER code, where 2k-1<=d<=n-1, when the set of helper nodes includes all remaining systematic nodes.

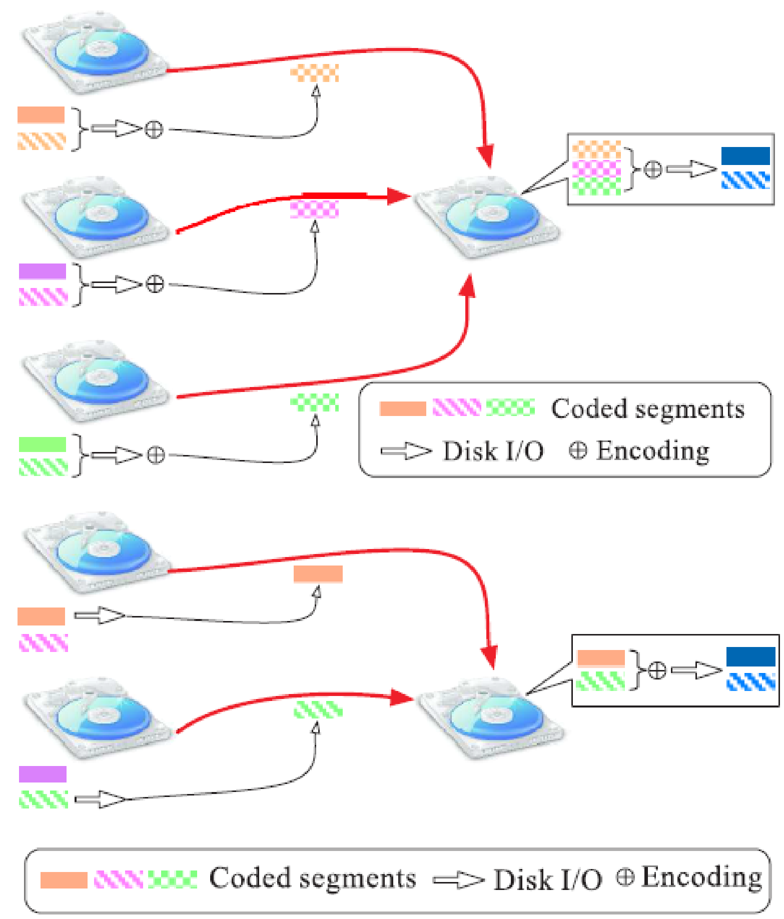
- When $\beta = 1$, there does not exist an exact [n,k,d] MSR code for $d < 2k - 3$

Factors in DSS

- Repair Bandwidth
- Disk I/O Read
- Storage Efficiency
- Repair Degree
- Repair Delay

Factors in DSS

- Repair Bandwidth
- Disk I/O Read
- Storage Efficiency
- Repair Degree
- Repair Delay

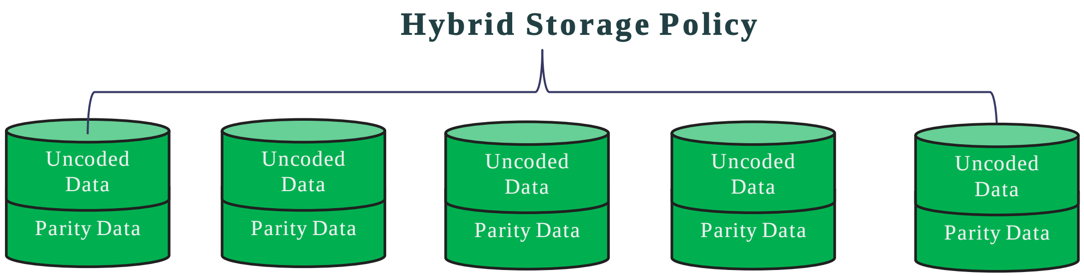


1. Increase disk life
2. Reduce repair delay

Hybrid Storage Policy(HSP)

Factor of HSP

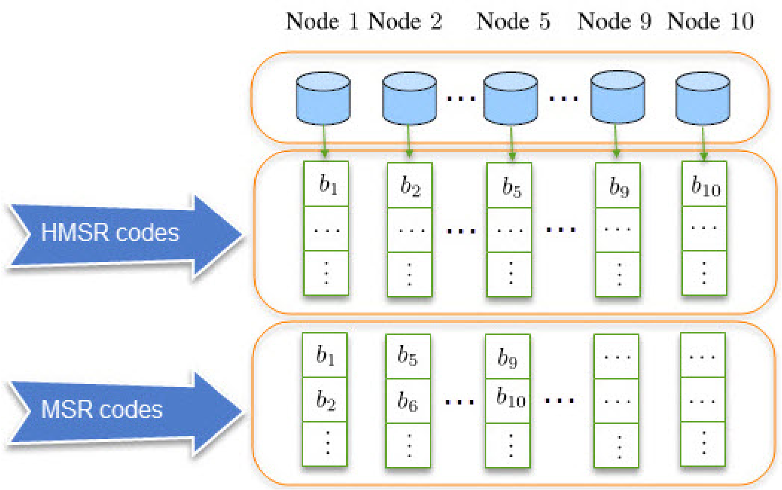
- Maximum concurrency ability for data access
- Approximate maximum Storage efficiency if data encoded by MSR codes
- Minimum repair degree
- Optimal repair bandwidth if using MSR codes



Hybrid Storage Policy(HSP)

Factor of HSP

- Maximum concurrency ability for data access
- Approximate maximum Storage efficiency if data encoded by MSR codes
- Minimum repair degree
- Optimal repair bandwidth if using MSR codes



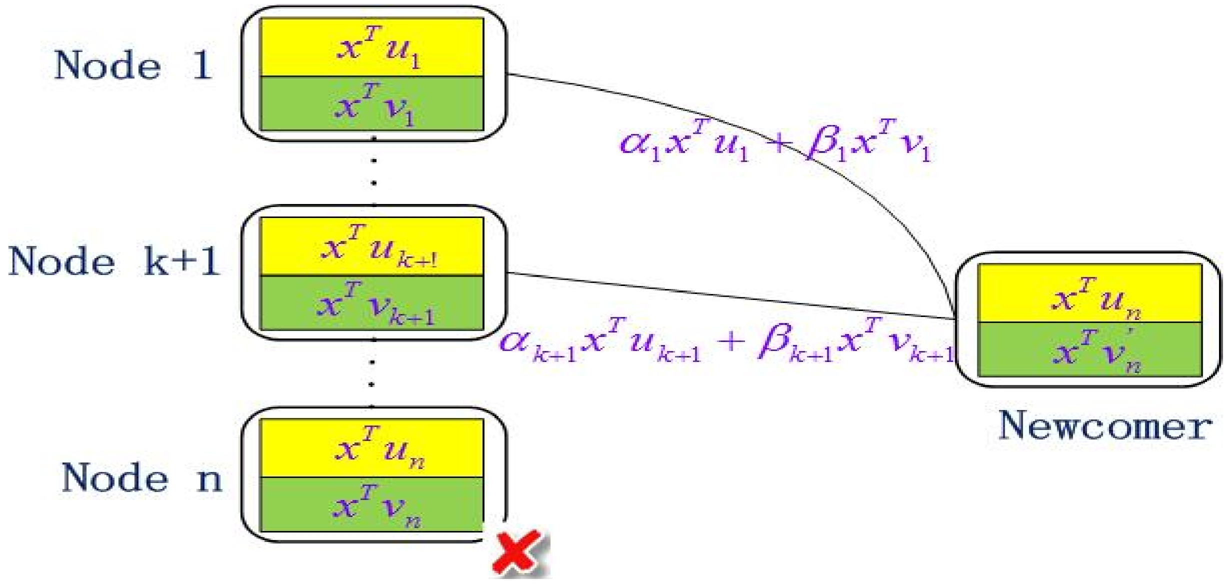
Assume the size of file is M = $500Mb$, which is divided into 10 bulks ($[b_1,\cdots, b_{10}]$), each size is $50Mb$, let the transfer rate be $10Mb/s$, then

Repair Bandwidth:

HSP: 5/s

Traditional:10/s(Maximum)

Hybrid Repair and Construction[4]



$\{u_i, v_i\}, i \in [n], is\ a\ (2n, 2k)\ MDS\ code.$

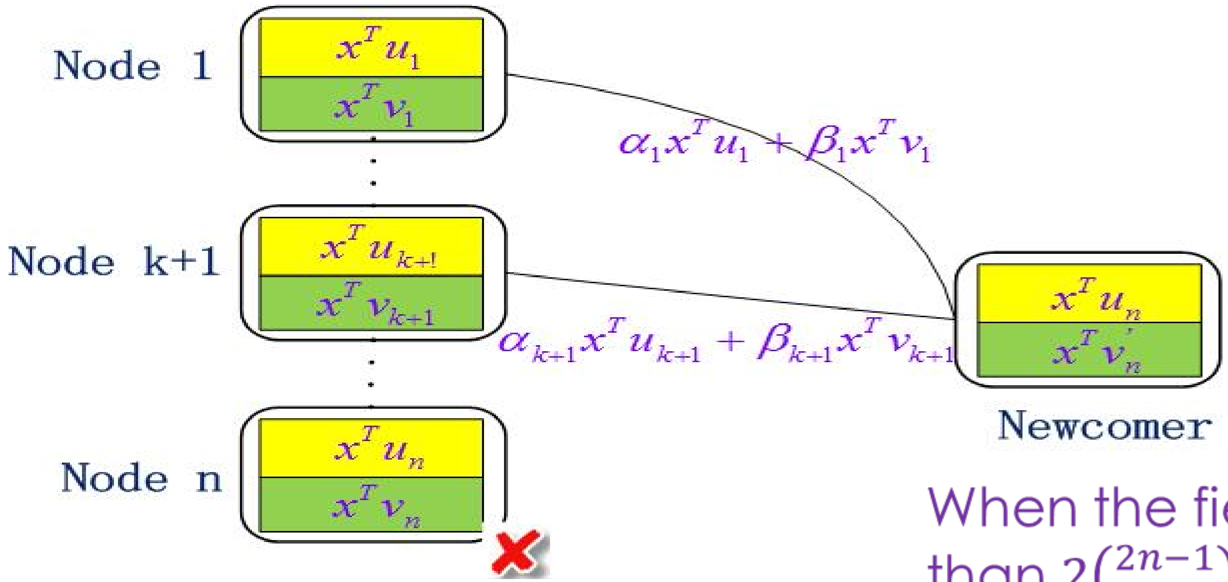
$$\sum_{i=1}^{k+1} (\alpha_i x^T u_i + \beta_i x^T v_i) = x^T u_n$$

$X \in F^{2k}$, *the original* message *vector*.

$$\sum_{i=1}^{k+1} \rho_i (\alpha_i x^T u_i + \beta_i x^T v_i) = x^T v_n'$$

[4] Wu Y. A construction of systematic MDS codes with minimum repair bandwidth[J]. Information Theory, IEEE Transactions on, 2011, 57(6):3738-3741

Hybrid Repair and Construction[4]

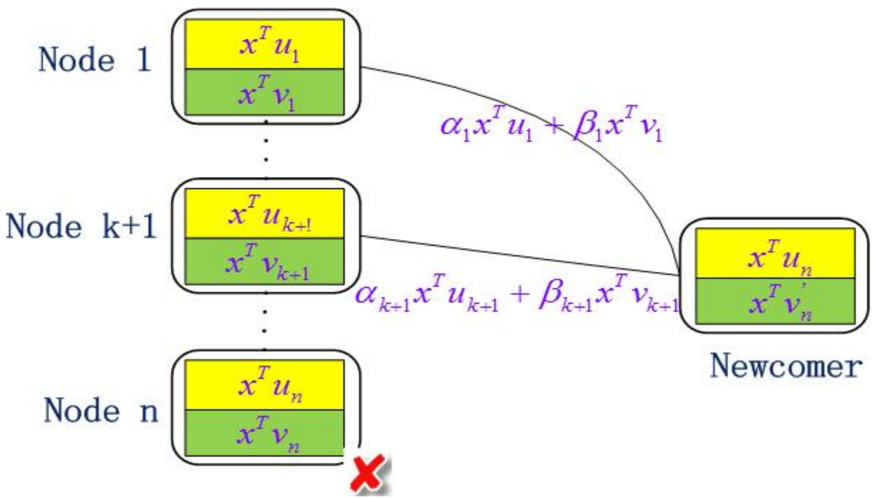


When the field size is greater than $2\binom{2n-1}{2k-1}$, there is an assignment of variables $\{\alpha_i, \beta_i, \rho_i\}$ satisfying the following repair formula. [5]

$$\sum_{i=1}^{k+1} (\alpha_i x^T u_i + \beta_i x^T v_i) = x^T u_n$$

$$\sum_{i=1}^{k+1} \rho_i (\alpha_i x^T u_i + \beta_i x^T v_i) = x^T v_n'$$

Hybrid Repair and Construction[4]



Strengths

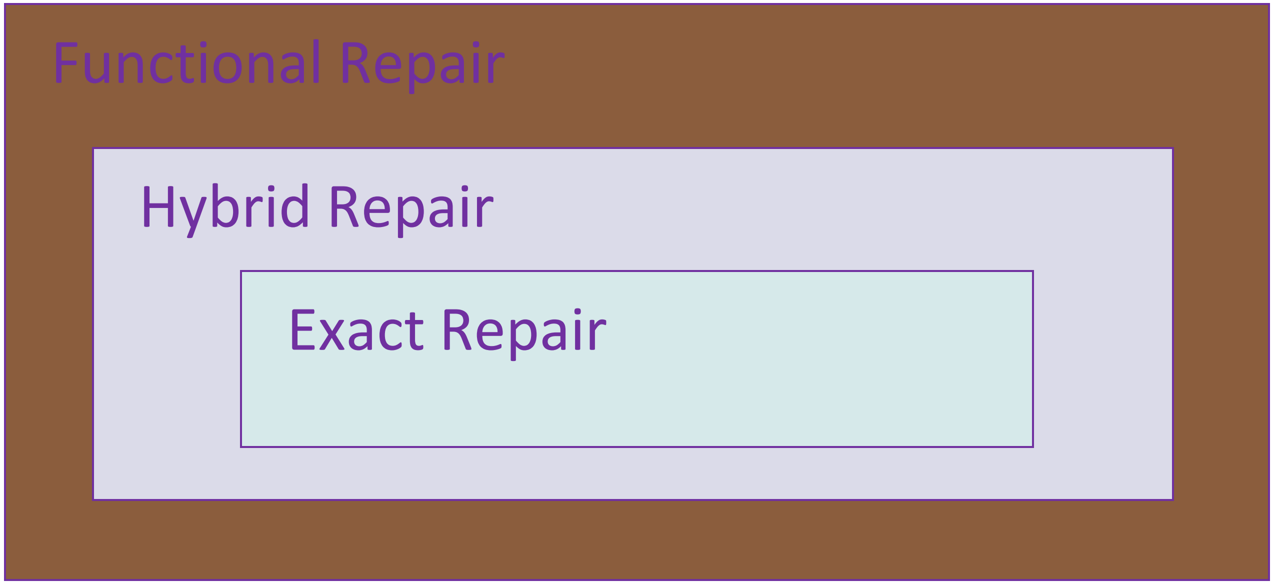
- A General Construction
- Inherit advantage of Hybrid Storage Policy

weakness

- Large encoding field, high complexity of repair algorithm.
- Can not use traditional decoding algorithm after several times of nodes repairing.

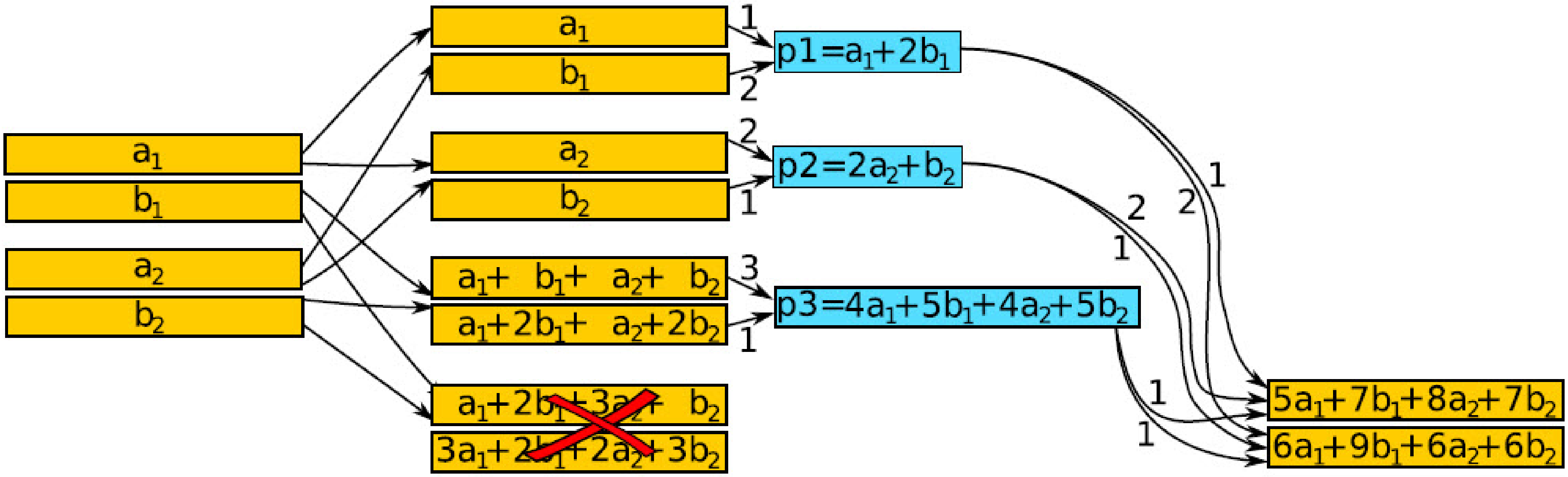
Repair Model

- Functional Repair
- Hybrid Repair
- Exact Repair



Functional Repair

Functional Repair Example:



Problems

Question

By hybrid storage policy, do there exist exact-repair MSR codes for $\beta = 1$ and $\alpha = 2$? If we can find this code, what are the advantages compare with that under hybrid repair model?

$$\gamma = \frac{Bd}{k(d-k+1)} [4]$$

$$d = k - 1 + \alpha$$

When the distributed storage system is homogeneous

Let $\beta = 1$. There is no exact-repair MSR codes when $d < 2k - 3$

For $\beta = 1$ and $\alpha = 2$, there is no exact repair MSR code when $k \geq 5$,

Problems

Question

By hybrid storage policy, do there exist exact-repair MSR codes for $\beta = 1$ and $\alpha = 2$? If we can find this code, what are the advantages compare with that under hybrid repair model?

When $k \geq 5$, $\beta = 1$ and $\alpha = 2$, there is no exact-repair MSR code.

$k = 2$, [4, 2, 3] HMSR Code

$k = 3$, [6, 3, 4] HMSR Code

$k = 4$, [8, 4, 5] HMSR Code

Liang S, Yuan C, Kan H. Linear Exact-repair Construction of Hybrid MSR Codes in Distributed Storage Systems[J]. IEEE Communications Letters 18(7): 1095-1098 (2014)

Hybrid MSR(HMSR) Code

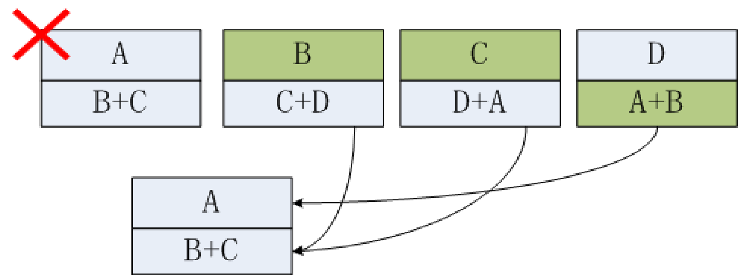
HMSR Code Under Exact Repair Model

- **$k = 2$, [4, 2, 3] HMSR Code**
- $k = 3$, [6, 3, 4] HMSR Code
- $k = 4$, [8, 4, 5] HMSR Code

Construction

node 1	node 2	node 3	node 4
A	B	C	D
$B+C$	$C+D$	$D+A$	$A+B$

Repair



Algorithm 1 A Repairing Algorithm for $[4, 2, 3]$-HMSR Codes

1: Download the first fragments from the next two nodes $i+1$ and $i+2$. Denote the symbols are d_1, d_2. The second fragments of node i can be repaired by $d_1 + d_2$. We set the next node of node 4 is node 1.

2: Download the second fragments from node $i+3$, which are denoted d_3, then the first fragment of node i can be repaired by $d_1 + d_3$.

Hybrid MSR(HMSR) Code

HMSR Code under Exact Repair Model

$k=2$, [4, 2, 3] HMSR Code

$k=3$, [6, 3, 4] HMSR Code

$k=4$, [8, 4, 5] HMSR Code

Construction

$A+S$	$B+S$	$C+S$	$D+S$	$E+S$	$F+S$
$B+C+S$	$C+D+S$	$D+E+S$	$E+F+S$	$F+A+S$	$A+B+S$
node 1	node 2	node 3	node 4	node 5	node 6

$$\text{where } S = A+B+C+D+E+F$$

Repair

We prove that any 4 out of 5 available nodes could repair the failed node with the minimum repair bandwidth.

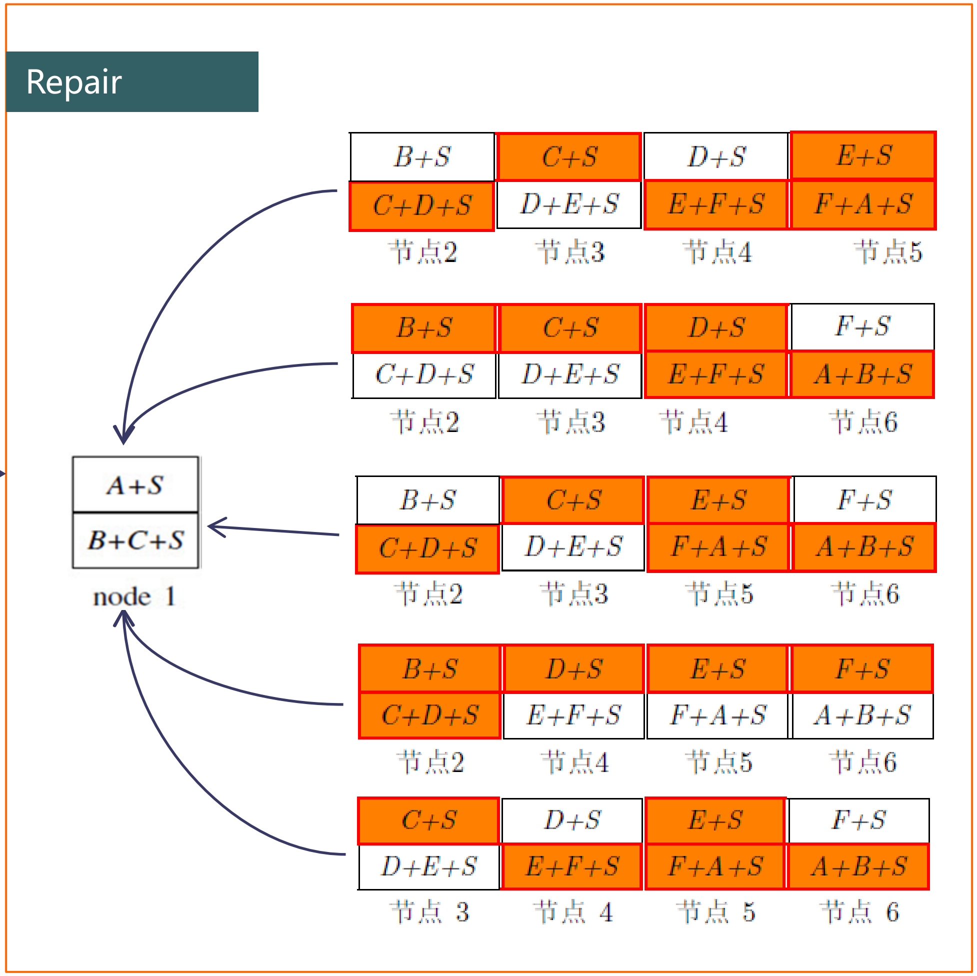
Hybrid MSR(HMSR) Code

HMSR Code Under Exact Repair Model

$k=2$, [4, 2, 3] HMSR Code

$k=3$, [6, 3, 4] HMSR Code

$k=4$, [8, 4, 5] HMSR Code



Hybrid MSR(HMSR) Code

HMSR Code Under Exact Repair Model

$k = 2$, [4, 2, 3] HMSR Code

$k = 3$, [6, 3, 4] HMSR Code

$k = 4$, [8, 4, 5] HMSR Code

Comparison

	Product Matrix[3]	ours
Required Finite Field	$F_q, q \geq 13$	F_2
Disk I/O Reading in Repairing Process	8	5
Concurrency Access	Good	Best
Disk I/O Reading for Repairing Two Failed nodes	8	6

Hybrid MSR(HMSR) Code

HMSR Code Under Exact Repair Model

$k=2$, [4, 2, 3] HMSR Code

$k=3$, [6, 3, 4] HMSR Code

$k=4$, [8, 4, 5] HMSR Code

$d=2k-3$ → Unknown

Hybrid MSR(HMSR) Code

Thank you !