Flexible Partial MDS Codes

Weiqi Li, Taiting Lu, Zhiying Wang, Hamid Jafarkhani

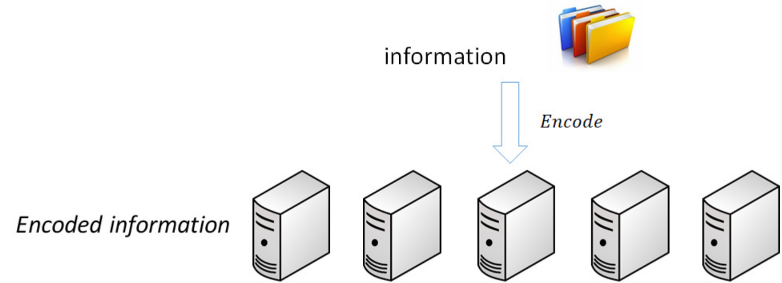
Center for Pervasive Communications and Computing (CPCC)
University of California, Irvine, USA
{weiqil4, taitingl, zhiying, hamidj}@uci.edu

Table of contents

Background and Contributions

Background

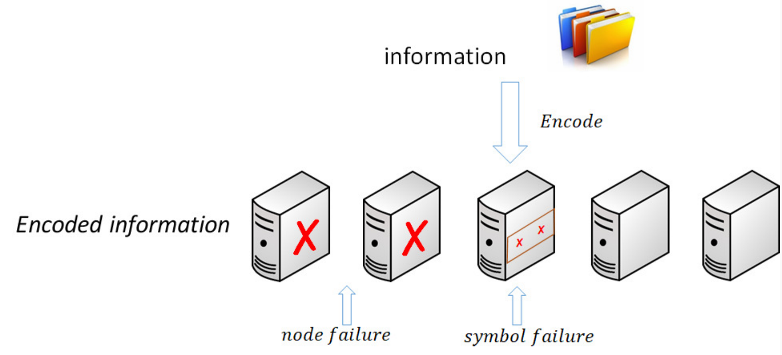
- Data storage is critical for non volatile memories.



- Number of storage nodes: n
- Reconstruction parameter: k

Background

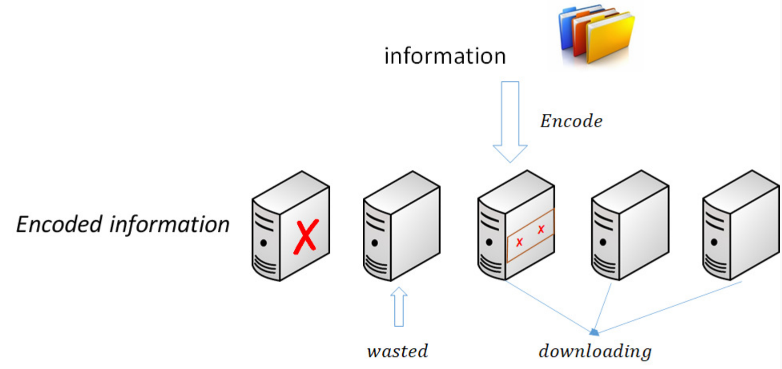
- PMDS codes [1] can tolerate node failures and additional symbol failures.



[1] M. Blaum, J. L. Hafner, and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures"

Background

- PMDS code can only make use of k nodes.
- The rest nodes are wasted.
- Each node downloading all symbols $\Rightarrow$ large latency.



Contributions

- We provide flexible PMDS code.

Table 1: An example of $(5, 3, 4, 2)$ flexible PMDS code with $\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\}$.

$C_{1,1,1}$	$*$	$C_{1,1,3}$	$*$	$*$
$C_{1,2,1}$	$*$	$C_{1,2,3}$	$C_{1,2,4}$	$*$
$C_{1,3,1}$	$*$	$*$	$C_{1,3,4}$	$*$
$C_{2,1,1}$	$*$	$C_{2,1,3}$	$C_{2,1,4}$	$*$

- tolerate up to 2 node failures and 2 symbols failures

Contributions

- With 1 node failure and 2 symbols failures, we download 3 symbols in each node.

Table 2: An example of $(5, 3, 4, 2)$ flexible PMDS code with $\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\}$.

$C_{1,1,1}$	$C_{1,1,2}$	$C_{1,1,3}$	$*$	$*$
$C_{1,2,1}$	$C_{1,2,2}$	$C_{1,2,3}$	$C_{1,2,4}$	$*$
$C_{1,3,1}$	$C_{1,3,2}$	$*$	$C_{1,3,4}$	$*$
$C_{2,1,1}$	$C_{2,1,2}$	$C_{2,1,3}$	$C_{2,1,4}$	$*$

Related work

- [Blaum-Hafiner-Hetzler, 2013], first work for PMDS codes.
- [Calis-Koyluoglu, 2016], general constructions for all parameters.
- [Gabrys-Yaakobi-Blaum-Siegel, 2019], constructions for PMDS codes over small fields.
- [Blaum, 2020], a hierarchical architecture that can tolerate different number of symbols failures in different layers.

Constructions

Main Idea

- Based on [Calis-Koyluoglu, 2016].
- Original information $\Rightarrow$ Gabidulin codeword symbols $\Rightarrow$ MDS code in each layer

$C_{1,1}$	$C_{1,2}$	$\cdots$	$\cdots$	C_{1,k_1}	parity
$C_{2,1}$	$C_{2,2}$	$\cdots$	C_{2,k_2}	parity	parity
$\vdots$	$\vdots$	$\ddots$	$\vdots$	parity	parity
$C_{a,1}$	$\cdots$	C_{a,k_a}	parity	parity	parity

- n: number of nodes. k: reconstruction parameter.
- ℓ: number of symbols in each node.
- $k\ell = k_1\ell_1 = k_2\ell_2 = ... = k_a\ell_a$.
- s: number of additional symbol failures.

Code Constructions

$$\begin{bmatrix} C_{1,1,1} & C_{1,1,2} & \cdots & C_{1,1,n} \\ C_{1,2,1} & C_{1,2,2} & \cdots & C_{1,2,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1,\ell_1,1} & C_{1,\ell_1,2} & \cdots & C_{1,\ell_1,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{a,1,1} & C_{a,1,2} & \cdots & C_{a,1,n} \\ C_{a,2,1} & C_{a,2,2} & \cdots & C_{a,2,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{a,\ell_a-\ell_{a-1},1} & C_{a,\ell_a-\ell_{a-1},2} & \cdots & C_{a,\ell_a-\ell_{a-1},n} \end{bmatrix}, \tag{1}$$

- Layer j: $(\ell_{j-1}+1)$-th row to ℓ_j-th row.
- $C_{j,m_j,i}$, j: layer index, m_j: row index, i: node index.

Encoding

- Original information $\rightarrow$ Gabidulin codeword symbols:
  - $K = k\ell - s$ original information symbols.
  - $N = \sum_{j=1}^{a} k_j(\ell_j - \ell_{j-1})$ Gabidulin codeword symbols.
- Gabidulin codeword symbols $\rightarrow$ MDS code in each layer:
  - (n, k_j) MDS code in each row.

$$[C_{j,m_j,k_j+1}, ..., C_{j,m_j,n}] = [C_{j,m_j,1}, ..., C_{j,m_j,k_j}]G_{n,k_j},$$

Encoding

Table 3: An example of $(5,3,4,2)$ flexible PMDS code with $\{(k_1,\ell_1),(k_2,\ell_2)\} = \{(4,3),(3,4)\}$.

$C_{1,1,1}$	$C_{1,1,2}$	$C_{1,1,3}$	$C_{1,1,4}$	$C_{1,1,5}$
$C_{1,2,1}$	$C_{1,2,2}$	$C_{1,2,3}$	$C_{1,2,4}$	$C_{1,2,5}$
$C_{1,3,1}$	$C_{1,3,2}$	$C_{1,3,3}$	$C_{1,3,4}$	$C_{1,3,5}$
$C_{2,1,1}$	$C_{2,1,2}$	$C_{2,1,3}$	$C_{2,1,4}$	$C_{2,1,5}$

- 10 information symbols $\rightarrow$ 15 Gabidulin codeword symbols.
- $(5,4)$ MDS code in row $1 \sim 3$. $(5,3)$ MDS code in row 4.

Decoding

- $n - k_J$ node failures, s symbol failures.
- MDS code in each layer $\rightarrow$ Gabidulin codeword symbols :
  - $t_{m_j} \leq k_J$ Gabidulin codeword symbols decoded in row m_j of layer j.
  - Totally K Gabidulin codeword symbols are decoded:

$$\sum_{j=1}^{J} \sum_{m_j=1}^{\ell_j - \ell_{j-1}} t_{m_j} = \ell_J k_J - s = K.$$

- Gabidulin codeword symbols $\rightarrow$ Original information:
  - Original information can be decoded from K Gabidulin codeword symbols.

Decoding

Table 4: An example of $(5, 3, 4, 2)$ flexible PMDS code with $\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\}$.

$C_{1,1,1}$	$C_{1,1,2}$	$C_{1,1,3}$	$*$	$*$
$C_{1,2,1}$	$C_{1,2,2}$	$C_{1,2,3}$	$C_{1,2,4}$	$*$
$C_{1,3,1}$	$C_{1,3,2}$	$*$	$C_{1,3,4}$	$*$
$C_{2,1,1}$	$C_{2,1,2}$	$C_{2,1,3}$	$C_{2,1,4}$	$*$

- 1 node failure, row $1 \sim 3$ are read in each node.
- 10 Gabidulin codeword symbols can be decoded.

Decoding

Table 5: An example of $(5, 3, 4, 2)$ flexible PMDS code with $\{(k_1, \ell_1), (k_2, \ell_2)\} = \{(4, 3), (3, 4)\}$.

$C_{1,1,1}$	$*$	$C_{1,1,3}$	$*$	$*$
$C_{1,2,1}$	$*$	$C_{1,2,3}$	$C_{1,2,4}$	$*$
$C_{1,3,1}$	$*$	$*$	$C_{1,3,4}$	$*$
$C_{2,1,1}$	$*$	$C_{2,1,3}$	$C_{2,1,4}$	$*$

- 2 node failures, all symbols are read in each node.
- 10 Gabidulin codeword symbols can be decoded.

Performance

- Assume 15 nodes, each node has failure probability $p = 0.2$, download 1 symbol takes time t.

	Fixed $(k, \ell) = (8, 15)$	Fixed $(k, \ell) = (10, 12)$	Fixed $(k, \ell) = (12, 10)$	Flexible $\{(k_1, \ell_1), (k_2, \ell_2), (k_3, \ell_3)\} = \{(12, 10), (10, 12), (8, 15)\}$
Probability of success	99.58%	93.89%	64.82%	99.58%
Average latency	$15t$	$12t$	$10t$	$10.82t$

Extensions

Extensions

- Similar construction can be applied to MDS codes, with optimal repair meeting MSR bound.
- LRC codes are also applied to the construction, with locality and flexible recovering nodes.