MMC, Sep 7,2017

MDS Codes for Distributed Storage System

Speaker	Xiaohu Tang
Email	xhutang@swjtu.edu.cn

Joint works with J. Li, P. Udaya, and C. Tian

Outline

1. Distributed storage system
2. Regenerating code
3. Our works

The age of big data

Jim Gray
1998 Turing Award Winner

Every 18 months

New storage=Sum of all old storage

Big data

IDC reported the size of the digital universe exceeded

- 1 ZB in 2010
- 1.8 Zb in 2011
- 35 Zb expected in 2020

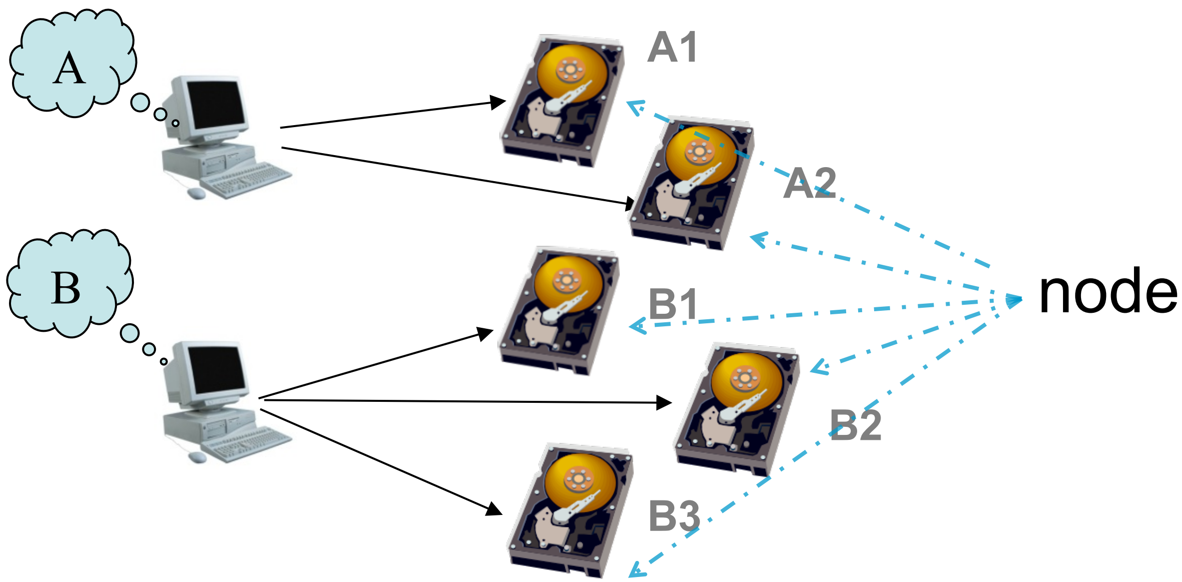
1ZB=1024EB 1EB=1024PB
1PB=1024TB 1TB=1024GB
1 ZB≈10^(12) GB!

Challenge

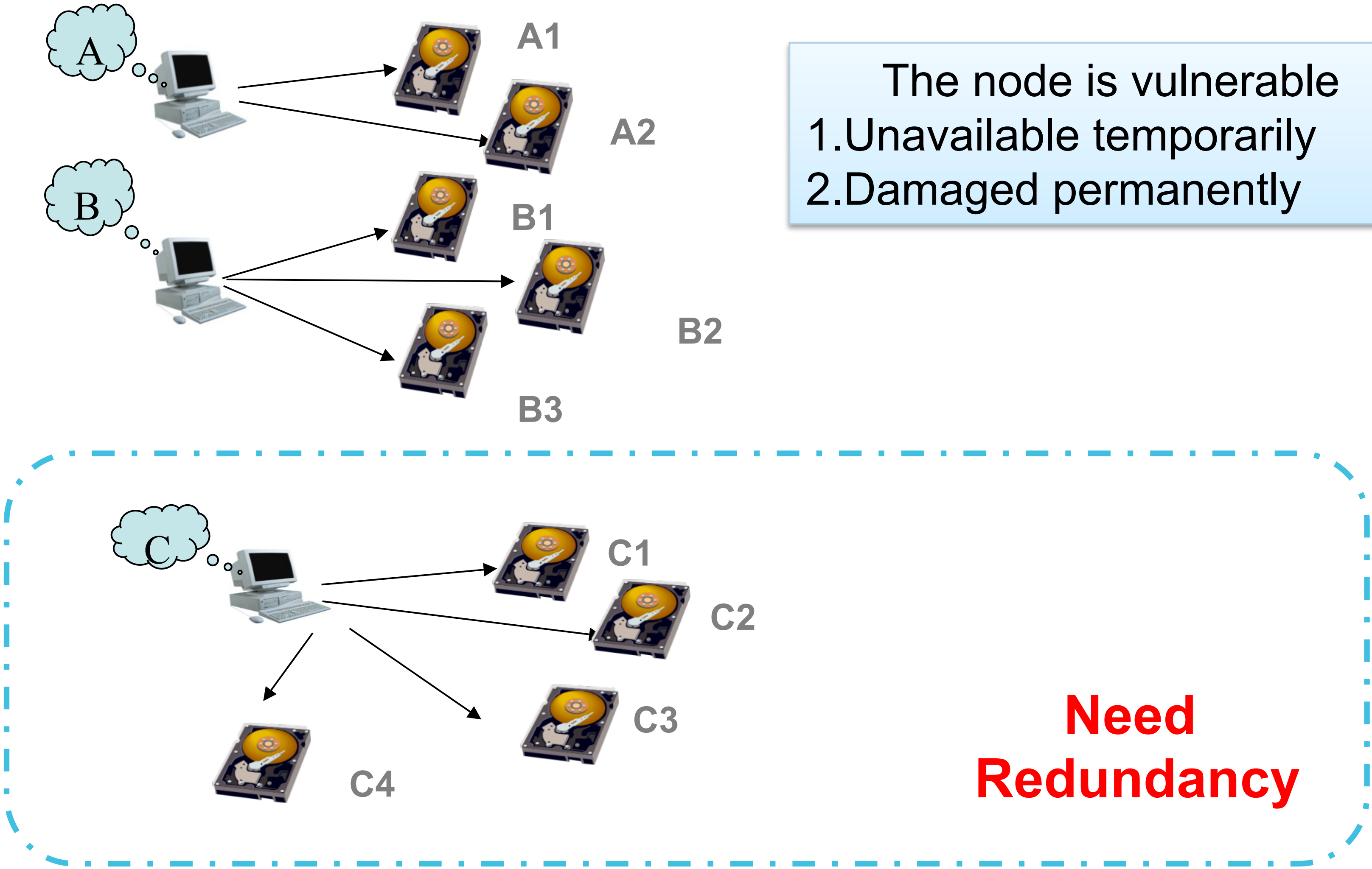
How to store big data?

Solutions: Centralized VS Distributed

Centralized storage	Distributed Storage
• Specific sever • Specific disk array • Bad scalability • Expensive	• Multiple independent device • Good scalability • Cheap



Reliability

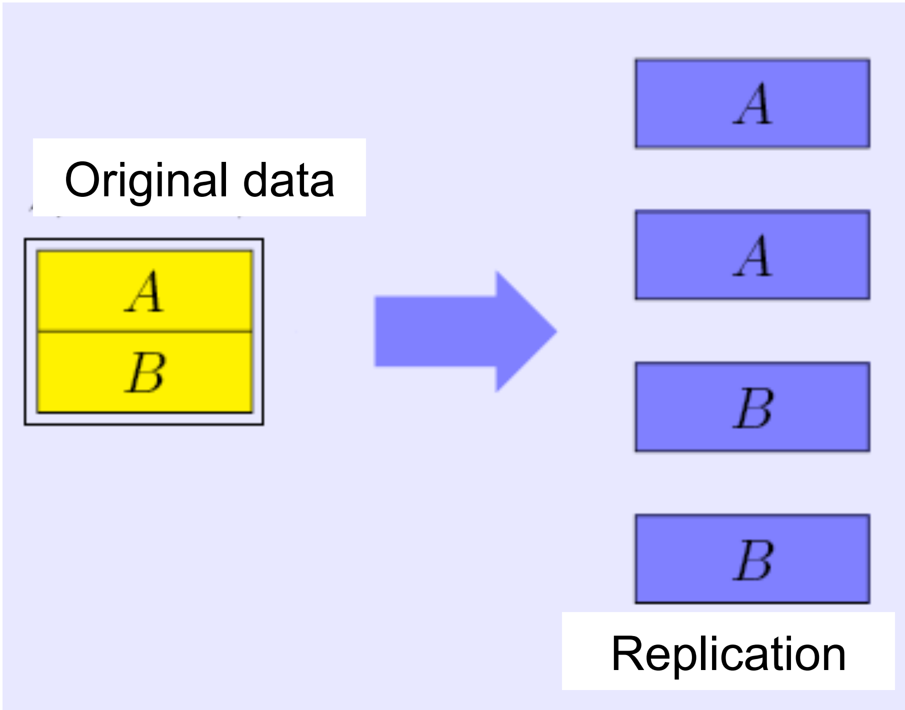


The node is vulnerable
1.Unavailable temporarily
2.Damaged permanently

Need Redundancy

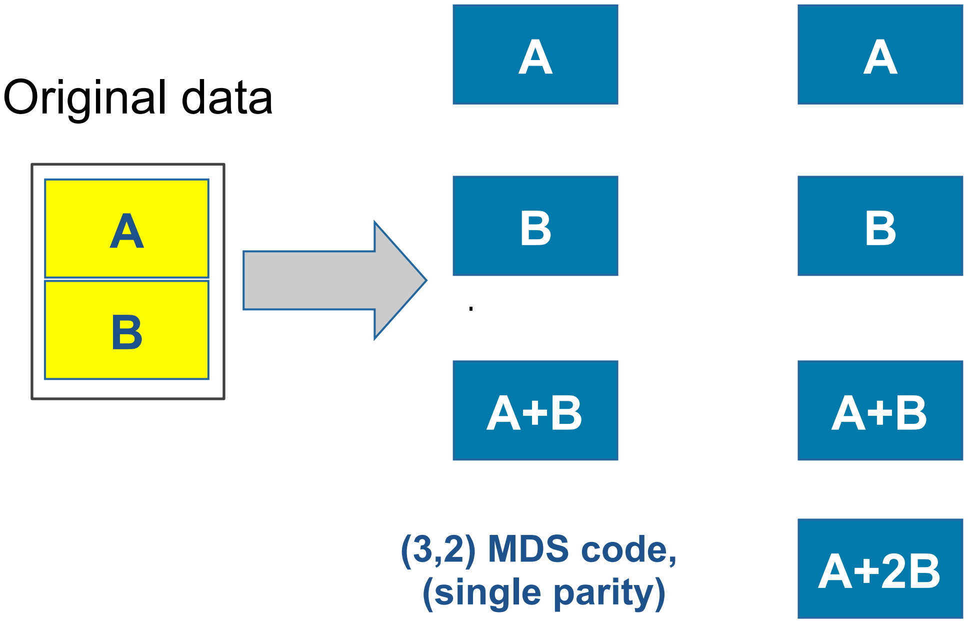
Two mechanisms for redundancy

- Replication



Two mechanisms for redundancy

- Erasure Code



(3,2) MDS code, (single parity) used in RAID 5

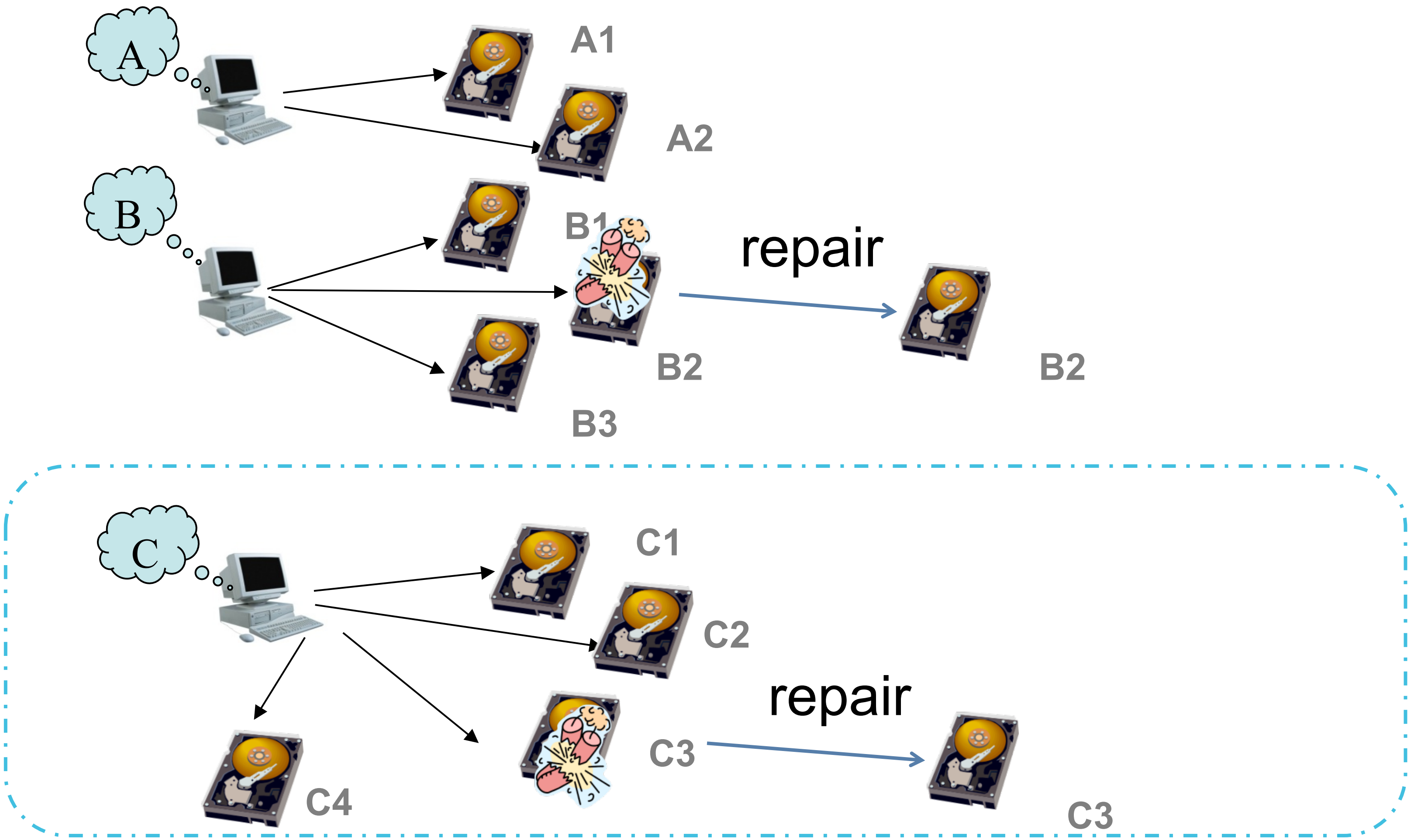
(4,2) MDS code. Tolerates any 2 failures

Used in RAID 6

Advantage:
- Lower redundancy or
- Higher reliability

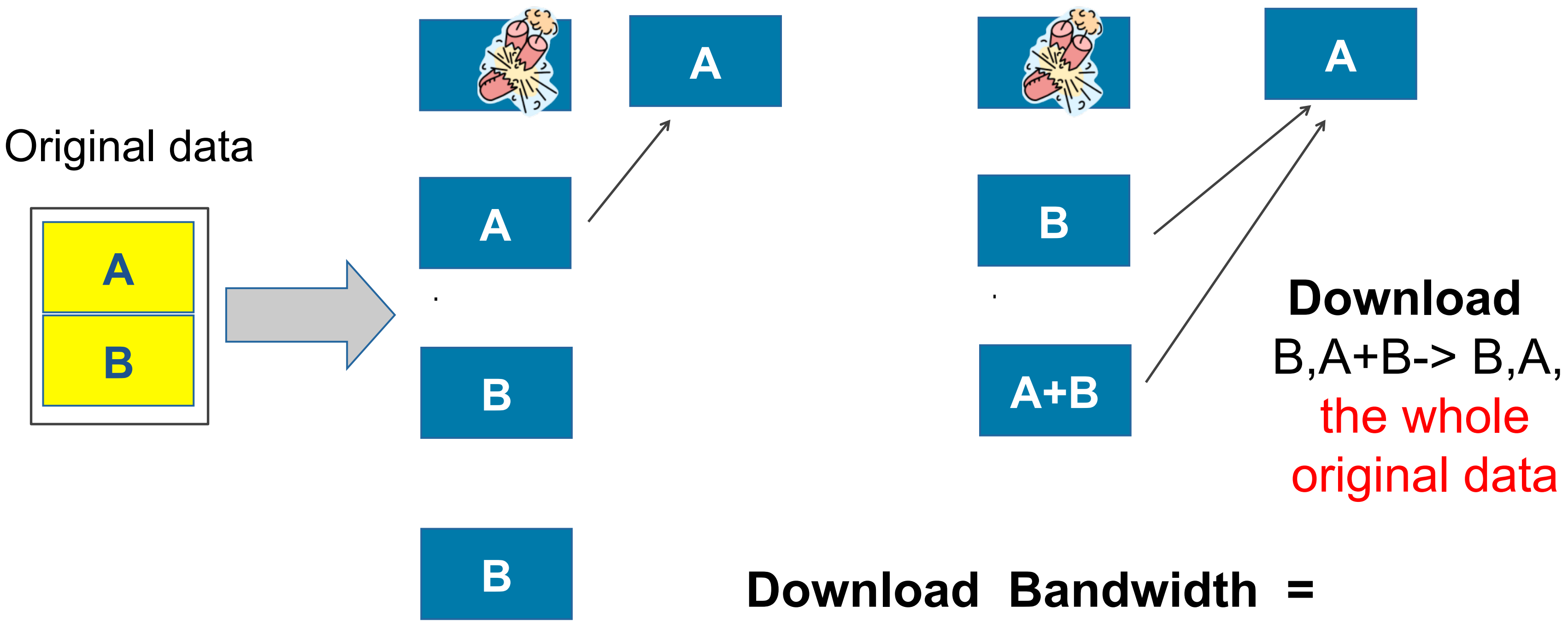
Repair

- Maintain Redundancy



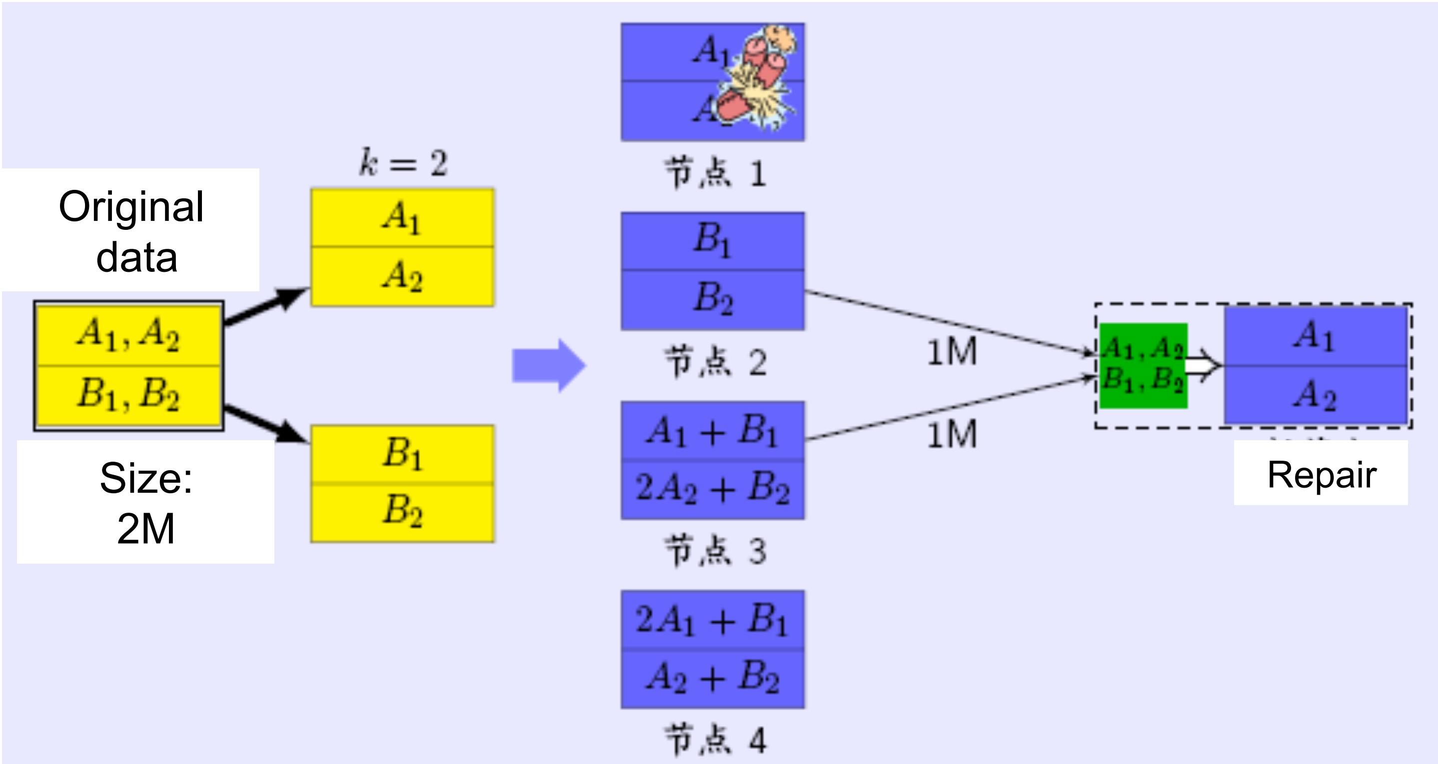
Repair

Replication VS Erasure Code



Download Bandwidth = Amount of download data for repairing one node

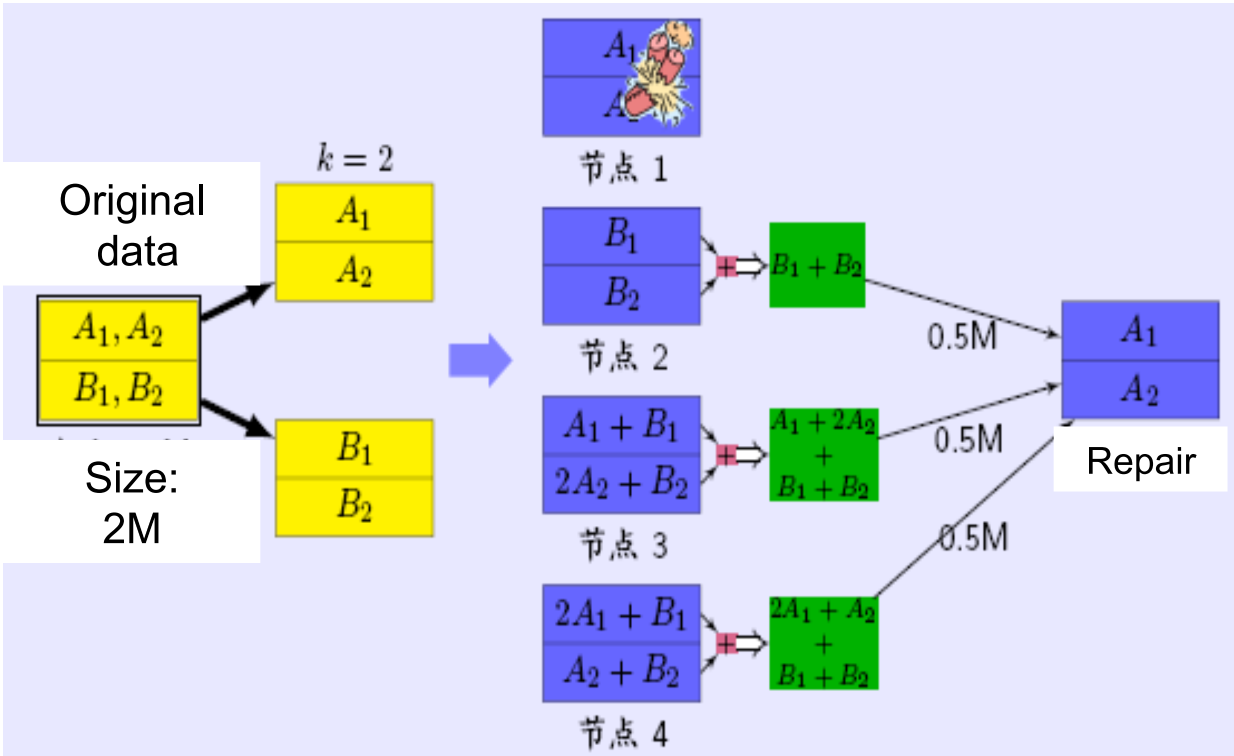
Erasure code



Download bandwidth 2M

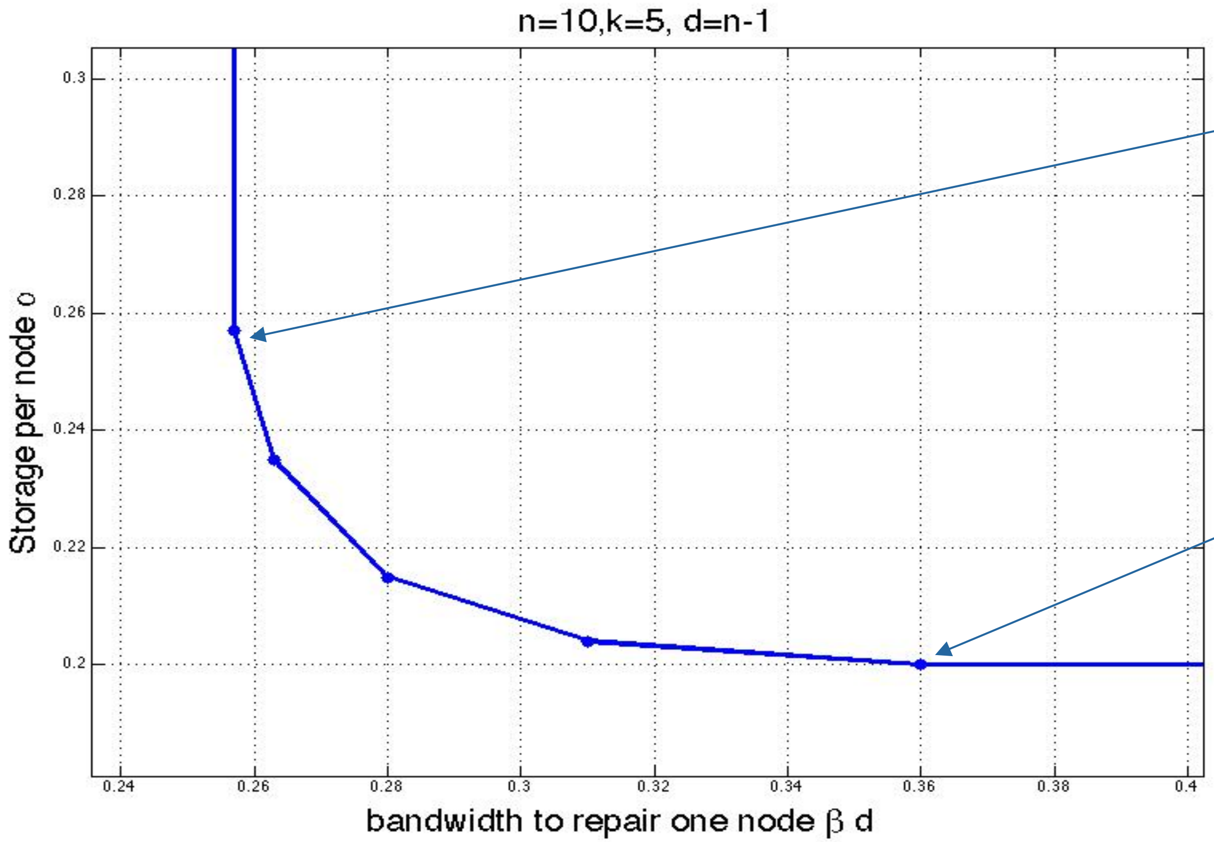
Regenerating code

2007 A. G. Dimakis *et al.*



Download bandwidth 1.5M

Storage-Communication tradeoff



Min-Bandwidth Regenerating code

Min-Storage Regenerating code

A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.*

State of the art

Before 2014

Optimal repair	Rate	
	≤0.5	>0.5
Systematic node	Completely	Partially
Parity node		Seldom

Rate=The size of original file/The storage

Product matrix method

- MBR for any possible parameters

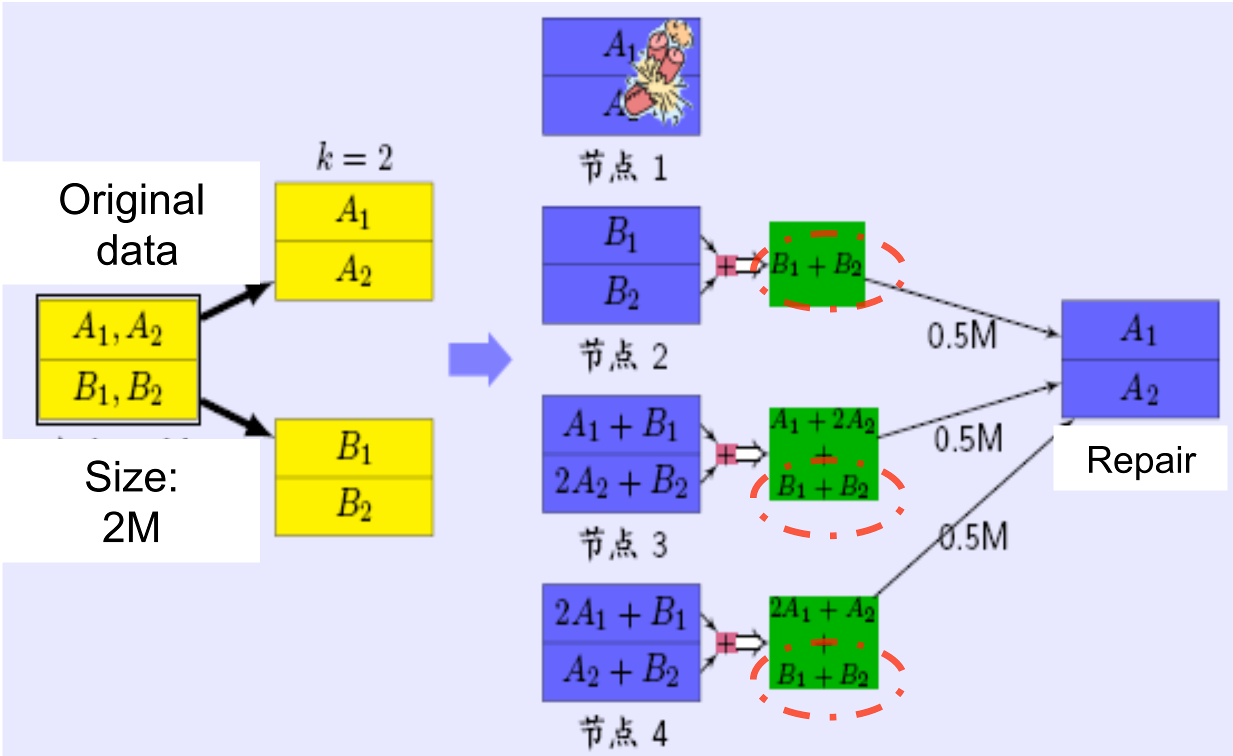
- MSR for

$$(n,k,\ d \geq 2k-2)$$

Rate<1/2

K.V. Rashmi, N.B. Shah, and P.V. Kumar, Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction, IEEE Trans. Inf. Theory, Vol. 57, N0. 8, pp. 5227-5239, 2011

Interference alignment technique



Interference alignment: 3 equations but 4 unknowns

General case (n=k+r,k,d)

Systematic	Node 0	f_0
	⋮	⋮
	Node k-1	f_1
Parity	Node k	$g_0=A_{0,0}f_0+\ldots+A_{0,k-1}f_{k-1}$
	⋮	⋮
	Node k+r-1	$g_{r-1}=A_{r-1,0}f_0+\ldots+A_{r-1,k-1}f_{k-1}$

where f_i is a column vector of length α, $A_{i,j}$ is an square matrix of order α

rate: $k/(k+r)$

Most interesting Case (n=k+2,k,d=k+1)

- Data node 1: f_1
- Data node 2: f_2
- …
- Data node *k:* f_k
- Parity node 1:

$$f_1 + f_2 + \cdots + f_k$$

- Parity node 2:

$$A_1 f_1 + A_2 f_2 + \cdots + A_k f_k$$

where f_i is a column vector of length α, A_i is an square matrix of order α

Optimal repair

To repair node i, download half from other k+1 nodes by multiplying a matrix S_i of order $\alpha/2\times\alpha$

- Data node 1: $S_i f_1$
- Data node 2: $S_i f_2$
- …
- Data node k: $S_i f_k$
- Parity node 1:

$$S_i f_1 + S_i f_2 + \cdots + S_i f_k$$

- Parity node 2:

$$S_i A_1 f_1 + S_i A_2 f_2 + \cdots + S_i A_k f_k$$

Sufficient conditions

(k+1)* α/2 equations but $k*\alpha$ unknowns
- Solve α unknowns f_i
- Cancel (k − 1)α unknowns f_j, $j \neq i$

$$\mathrm{rank}\begin{pmatrix} S_i \\ S_i A_j \end{pmatrix} = \begin{cases} \frac{\alpha}{2}, & \text{if } i \neq j \\ \alpha, & \text{if } i = j \end{cases} \quad \text{for any } 1 \leq i, j \leq k.$$

Best known results

	k	α	Alphabet size
Zigzag	$m+1$	2^m	3
Long MDS	$3m$	2^m	2m+1

- T. Tamo, Z. Wang and J. Bruck, ``Zigzag codes: MDS array codes with optimal rebuilding," IEEE Trans. Inform. Theory, vol. 59, no. 3, pp. 1597-1616, Mar. 2013.
- Z. Wang, T. Tamo and J. Bruck, ``Long MDS codes for optimal repair bandwidth," Tech. Rep. Available at http :/ /paradise.caltech.edu/etr.html

Zigzag code

	0	1	2	R	Z
0	$\clubsuit$	$\spadesuit$	$\heartsuit$		$\clubsuit$
1	$\heartsuit$	$\diamondsuit$	$\clubsuit$		$\heartsuit$
2	$\spadesuit$	$\clubsuit$	$\diamondsuit$		$\spadesuit$
3	$\diamondsuit$	$\heartsuit$	$\spadesuit$		$\diamondsuit$

Properties

	0	1	2	R	Z
0	♣	♠	♡		♣
1	♡	◇	♣		♡
2	♠	♣	◇		♠
3	◇	♡	♠		◇

- **Optimal access property**

  Directly download without any computation

- **Optimal update property**

  Update only 2 bits in parity nodes when update one data, which is the minimal update

Our work

- Code construction with
  - Optimal access property
  - Optimal update property
- Optimal repair of parity nodes

Code construction

- **Includes the Zigzag codes and long MDS codes**

  Establish a general but simple framework of (k+2,k,k+1) MSR code based on invariant subspace technique, which unifies the best known cases

- **New constructions**

  Construct more MSR codes, some of which improve Zigzag

	New code $\mathcal{C}_1$	New code $\mathcal{C}_2$	New code $\mathcal{C}_3$	New code $\mathcal{C}_4$	The Zigzag code [18]	The Long MDS code [20]
k	$3m$	$2m$	$2m$	$2m$	$m+1$	$3m$
k_A	m	m	m	0	$m+1$	$2m$
k_U	m	m	$2m$	$2m$	$m+1$	m
$k_{A\&U}$	m	m	m	0	$m+1$	0
q	$\geq 2m+1$	$\geq m+1$	$\geq 2m+1$	$\geq m+1$	3	$\geq 2m+1$

J. Li, X.H. Tang, and U. Parampalli, A Framework of Constructions of Minimal Storage Regenerating Codes With the Optimal Access/Update Property, IEEE Trans. Inf. Theory, 61(4): 1920-1932 (2015)

Invariant subspace

Definition: Let q be a prime power and A *be a* $\alpha\times\alpha$ matrix. Assume that U is a subspace of F_q^{α} with $\dim(U)=s<\alpha$. Then U is said to be a invariant subspace with respect to A if

$$Aw \in U \text{ for any } w \in U$$

Definition: Let S be a matrix. Span(S) is defined as the vector space spanned by its rows.

$$\operatorname{rank}\begin{pmatrix} S_i \\ S_i A_j \end{pmatrix} = \begin{cases} \frac{\alpha}{2}, & \text{if } i \neq j \\ \alpha, & \text{if } i = j \end{cases} \quad \text{for any } 1 \leq i, j \leq k.$$

Invariant subspace

Assume that e_0 and e_1 are two arbitrary linearly independent row vectors of length α over F_q. Let

$$S = \begin{pmatrix} e_0 \\ e_1 \end{pmatrix}$$

Then Span(S) is an invariant subspace with respect to A if and only if

$$\begin{pmatrix} e_0 \\ e_1 \end{pmatrix} A = \begin{pmatrix} ae_0 + be_1 \\ ce_0 + de_1 \end{pmatrix} \text{ and } ad \neq bc,\ a, b, c, d \in \mathbf{F}_q$$

Invariant subspace

In details, there are 7 cases

1. $b=c=0$ and $a,d \neq 0$
2. $a=d=0$ and $b,c \neq 0$
3. $b=0$ and $a,b,c \neq 0$
4. $c=0$ and $a,b,d \neq 0$
5. $a=0$ and $b,c,d \neq 0$
6. $d=0$ and $a,b,c \neq 0$
7. ~~$a,b,c,d \neq 0$ and $ad \neq bc$~~

Equivalent

$\boxed{e_0 \leftrightarrow e_1}$

Invariant subspace

- type I if $\begin{pmatrix} e_0 \\ e_1 \end{pmatrix} A = \begin{pmatrix} ae_0 \\ de_1 \end{pmatrix}$
- type II if $\begin{pmatrix} e_0 \\ e_1 \end{pmatrix} A = \begin{pmatrix} be_1 \\ ce_0 \end{pmatrix}$
- type III if $\begin{pmatrix} e_0 \\ e_1 \end{pmatrix} A = \begin{pmatrix} ae_0 \\ ce_0 + de_1 \end{pmatrix}$
- type IV if $\begin{pmatrix} e_0 \\ e_1 \end{pmatrix} A = \begin{pmatrix} be_1 \\ ce_0 + de_1 \end{pmatrix}$

Our methods

Let $V = F_q^{\alpha}$, V_0 and V_1 be a partition of V with $|V_0|=|V_1|$. For simplicity, we still use V_0 (V_1) to denote the matrix formed by the rows of V_0 (V_1) .

Then A can be characterized by

$$\begin{pmatrix} V_0 \\ V_1 \end{pmatrix} A = \begin{pmatrix} aV_0 + bV_1 \\ cV_0 + dV_1 \end{pmatrix}$$

Goal: Find k such partitions $V_{i,0}$ and $V_{i,1}$ to determine the coding matrix A_i

Partition

Let $\alpha=2^m$, and e_i, $0\leq i<\alpha$ be a basis of F_q^{α} .

The *m* partitions are

$$\{e_0, e_1, \cdots, e_{2^m-1}\} = V_{1,0} \cup V_{1,1} = \cdots = V_{m,0} \cup V_{m,1}$$

such that

$$|V_{i_1,j_1} \cap V_{i_2,j_2} \cap \cdots \cap V_{i_l,j_l}| = 2^{m-l}$$

for any $1 \leq i_1 < i_2 < \cdots < i_l \leq m$, $j_t = 0, 1$, $1 \leq t \leq l \leq m$.

i	0	1
$V_{i,0}$	e_0 e_1	e_0 e_2
$V_{i,1}$	e_2 e_3	e_1 e_3

(a)

i	0	1	2	i	0	1	2
$V_{i,0}$	e_0 e_1 e_2 e_3	e_0 e_1 e_4 e_5	e_0 e_2 e_4 e_6	$V_{i,1}$	e_4 e_5 e_6 e_7	e_2 e_3 e_6 e_7	e_1 e_3 e_5 e_7

(b)

Our unified construction

Construction: The $(n = k+2, k, \alpha = 2^m)$ code $\mathcal{C}$ has coding matrix A_i of order $\alpha \times \alpha$ and repair matrix S_i of order $\frac{\alpha}{2} \times \alpha$ for $0 \le i < k$, such that

1. $\begin{pmatrix} V_{i,0} \\ V_{i,1} \end{pmatrix} A_i = \begin{pmatrix} a_i V_{i,0} + b_i V_{i,1} \\ c_i V_{i,0} + d_i V_{i,1} \end{pmatrix}$,
2. $S_i = u_i V_{i,0} + w_i V_{i,1}$,

where a_i, b_i, c_i, d_i, u_i and w_i can be coefficients in $\mathbf{F}_q$ or diagonal matrices over $\mathbf{F}_q$ such that

$$\begin{pmatrix} a_i V_{i,0} + b_i V_{i,1} \\ c_i V_{i,0} + d_i V_{i,1} \end{pmatrix}$$

is nonsingular.

Re-interpretation of Zigzag code

Construction of Zigzag: The $(n = k+2, k = m, \alpha = 2^m)$ Zigzag code $\mathcal{C}$ has coding matrix A_i of order $\alpha \times \alpha$ and repair matrix S_i of order $\frac{\alpha}{2} \times \alpha$ for $0 \le i < m$, such that

$$\boxed{\begin{pmatrix} V_{i,0} \\ V_{i,1} \end{pmatrix} A_i = \begin{pmatrix} b_i V_{i,1} \\ c_i V_{i,0} \end{pmatrix}} \quad \text{Type 2}$$

and

$$S_i = V_{i,0}$$

where b_i and c_i can be coefficients or diagonal matrices over $\{1, 2\}$.

Re-interpretation of long MDS code

Construction of long MDS: The $(n = k+2, k = 3m, \alpha = 2^m)$ MDS code $\mathcal{C}$ has coding matrix A_i of order $\alpha \times \alpha$ and repair matrix S_i of order $\frac{\alpha}{2} \times \alpha$ for $0 \le i < 3m$, such that

$$\begin{pmatrix} V_{i,0} \\ V_{i,1} \end{pmatrix} A_i = \begin{cases} \begin{pmatrix} a_i V_{i,0} + b_i V_{i,1} \\ d_i V_{i,1} \end{pmatrix}, & 0 \le i < m \\ \begin{pmatrix} a_i V_{i,0} \\ c_i V_{i,0} + d_i V_{i,1} \end{pmatrix}, & m \le i < 2m \\ \begin{pmatrix} a_i V_{i,0} \\ d_i V_{i,1} \end{pmatrix}, & 2m \le i < 3m \end{cases}$$

Type 3

Type 1

and

$$S_i = \begin{cases} V_{i,0} & 0 \le i < m \\ V_{i,1} & m \le i < 2m \\ V_{i,0} + w_i V_{i,1} & 2m \le i < 3m \end{cases}$$

Construction of new code 1

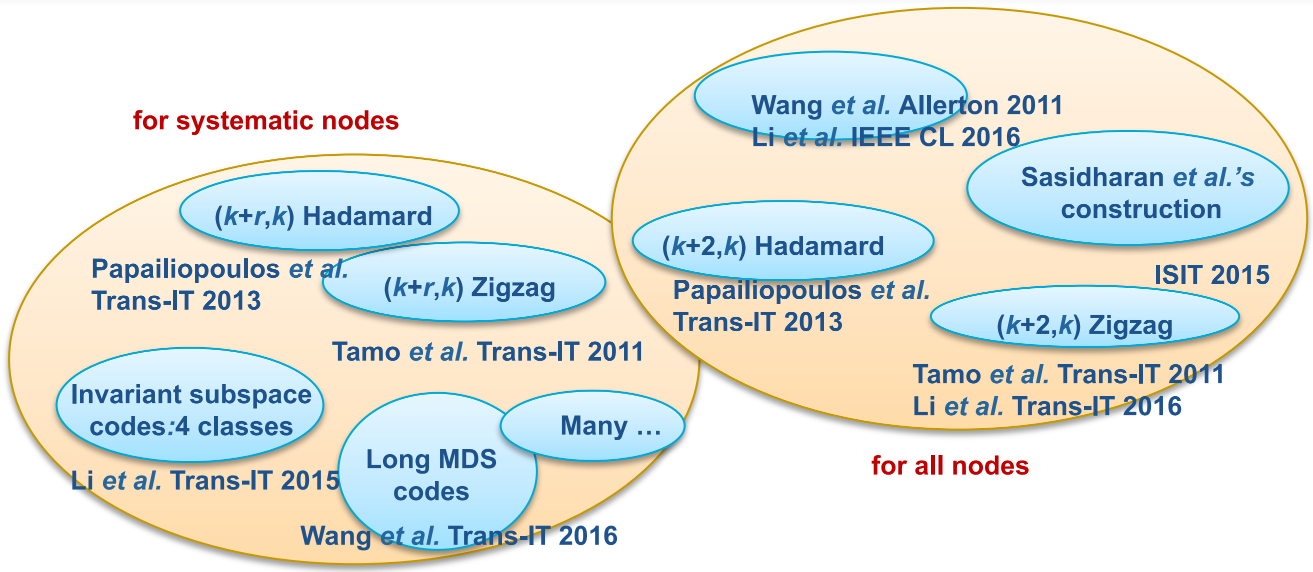
Construction 1. *The* $(n = k+2, k = 3m)$ *code* $\mathcal{C}_1$ *has coding matrices* A_i *of order* $\alpha \times \alpha$ *and repair matrices* S_i *of order* $\frac{\alpha}{2} \times \alpha$ *for* $1 \leq i \leq k$*, such that*

1. $$\begin{pmatrix} V_{i,0} \\ V_{i,1} \end{pmatrix} A_i = \begin{cases} \begin{pmatrix} \lambda_{i,1} V_{i,1} \\ \lambda_{i,0} V_{i,0} \end{pmatrix}, & 1 \leq i \leq m, \quad \text{Type 2} \\ \begin{pmatrix} \lambda_{i,0} V_{i,0} \\ \lambda_{i,1} V_{i,1} + k_{i-m} V_{i,0} \end{pmatrix}, & m+1 \leq i \leq 3m, \quad \text{Type 3} \end{cases}$$

2. $$S_i = \begin{cases} V_{i,0}, & \text{if } 1 \leq i \leq m, \\ V_{i,0} + t_{i-m} V_{i,1}, & \text{if } m+1 \leq i \leq 3m, \end{cases}$$

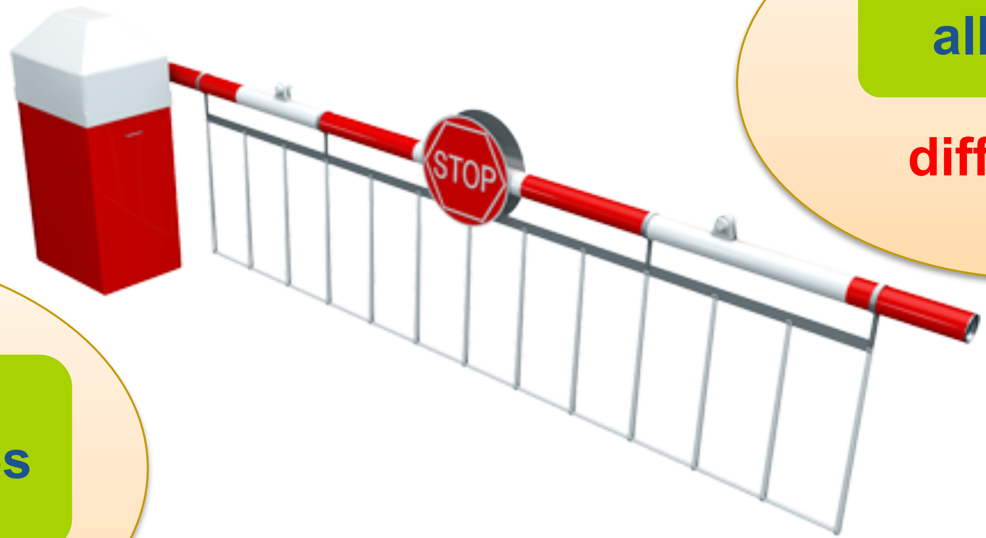
where $\lambda_{i,0}, \lambda_{i,1}, k_j, t_j \in \mathbf{F}_q^*$ *for all* $1 \leq i \leq k$ *and* $1 \leq j \leq 2m$.

Repair for parity nodes of high-rate code



1. Li, Tang and Tian, Enabling All-Node-Repair in Minimum Storage Regenerating Codes, arXiv:1604.07671, April 2106. **(d=n-1)**
2. Ye and Barg, Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization, arXiv:1605.08630, May 2016. **(d≤n-1)**
3. Sasidharan, Vajha, and Kumar, An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair, arXiv:1607.07335, July 2016. **(d≤n-1)**

Barrier



all nodes

difficult

systematic nodes

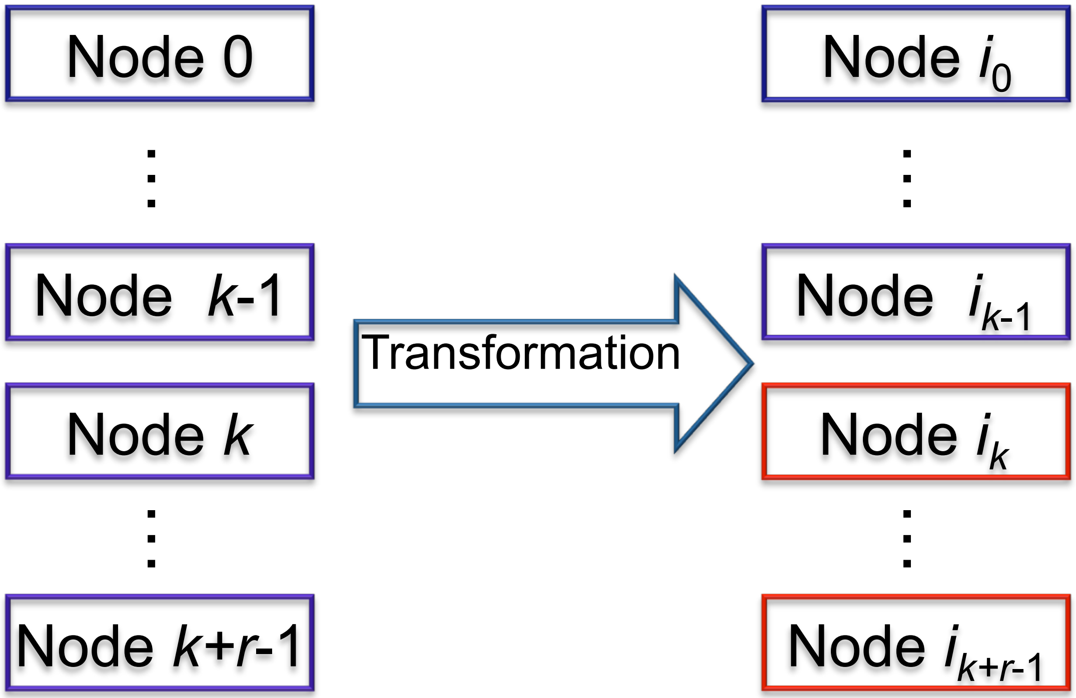
easy

Why is this happening?

A new transformation

Base MDS storage code

New MDS storage code



- ✓ ***r* nodes: optimal RB**
- ✓ ***k* nodes: same normalized RB**
- ✓ **same field size**
- ✓ **sub-packetization increased *r*-fold**

Procedure

Given a base MDS (storage) code

- Step 1: **Space sharing**
- Step 2: **Permuting**
- Step 3: **Paring**

Step 1

Space sharing r instances to get code C_1

		Instance 0	Instance 1	Instance 2
stationary nodes	Node 0	$f_0^{(0)}$	$f_0^{(1)}$	$f_0^{(2)}$
	Node 1	$f_1^{(0)}$	$f_1^{(1)}$	$f_1^{(2)}$
	Node 2	$f_2^{(0)}$	$f_2^{(1)}$	$f_2^{(2)}$
	Node 3	$f_3^{(0)}$	$f_3^{(1)}$	$f_3^{(2)}$
variable nodes	Node 4	$g_0^{(0)}$	$g_0^{(1)}$	$g_0^{(2)}$
	Node 5	$g_1^{(0)}$	$g_1^{(1)}$	$g_1^{(2)}$
	Node 6	$g_2^{(0)}$	$g_2^{(1)}$	$g_2^{(2)}$

Step 2

Permuting data in variable nodes of C_1 to get C_2

$$g_j^{(i)} \rightarrow g_{j+i}^{(i)}$$

variable nodes

	Instance 0	Instance 1	Instance 2
Node 4	$g_0^{(0)}$	$g_0^{(1)}$	$g_0^{(2)}$
Node 5	$g_1^{(0)}$	$g_1^{(1)}$	$g_1^{(2)}$
Node 6	$g_2^{(0)}$	$g_2^{(1)}$	$g_2^{(2)}$

C_2

$g_0^{(1)}$

In some cases, the permutations can be arbitrary.

Step 3

Paring data in variable nodes of C_2 to get C_3

$$(g_{i+j}^{(i)}, g_{i+j}^{(j)})$$

C_2

variable nodes

	Instance 0	Instance 1	Instance 2
Node 4	$g_0^{(0)}$	$g_1^{(1)}$	$g_2^{(2)}$
Node 5	$g_1^{(0)}$	$g_2^{(1)}$	$g_0^{(2)}$
Node 6	$g_2^{(0)}$	$g_0^{(1)}$	$g_1^{(2)}$

$\Downarrow$

C_3

variable nodes

	Instance 0	Instance 1	Instance 2
Node 4	$g_0^{(0)}$	$-g_1^{(1)}+g_1^{(0)}$	$-g_2^{(2)}+g_2^{(0)}$
Node 5	$g_1^{(0)}+g_1^{(1)}$	$g_2^{(1)}$	$-g_0^{(2)}+g_0^{(1)}$
Node 6	$g_2^{(0)}+g_2^{(2)}$	$g_0^{(1)}+g_0^{(2)}$	$g_1^{(2)}$

The resultant code

Structure of the MDS storage code C_3

		Instance 0	Instance 1	Instance 2
stationary nodes	Node 0	$f_0^{(0)}$	$f_0^{(1)}$	$f_0^{(2)}$
	Node 1	$f_1^{(0)}$	$f_1^{(1)}$	$f_1^{(2)}$
	Node 2	$f_2^{(0)}$	$f_2^{(1)}$	$f_2^{(2)}$
	Node 3	$f_3^{(0)}$	$f_3^{(1)}$	$f_3^{(2)}$
variable nodes	Node 4	$g_0^{(0)}$	$-g_1^{(1)}+g_1^{(0)}$	$-g_2^{(2)}+g_2^{(0)}$
	Node 5	$g_1^{(0)}+g_1^{(1)}$	$g_2^{(1)}$	$-g_0^{(2)}+g_0^{(1)}$
	Node 6	$g_2^{(0)}+g_2^{(2)}$	$g_0^{(1)}+g_0^{(2)}$	$g_1^{(2)}$

Optimal repair of variable nodes

Download column *i* to repair variable node *i*

		Instance 0	Instance 1	Instance 2
stationary nodes	Node 0	$f_0^{(0)}$	$f_0^{(1)}$	$f_0^{(2)}$
	Node 1	$f_1^{(0)}$	$f_1^{(1)}$	$f_1^{(2)}$
	Node 2	$f_2^{(0)}$	$f_2^{(1)}$	$f_2^{(2)}$
	Node 3	$f_3^{(0)}$	$f_3^{(1)}$	$f_3^{(2)}$
variable nodes	Node 4	$g_0^{(0)}$	$-g_1^{(1)}+g_1^{(0)}$	$-g_2^{(2)}+g_2^{(0)}$
	Node 5	$g_1^{(0)}+g_1^{(1)}$	$g_2^{(1)}$	$-g_0^{(2)}+g_0^{(1)}$
	Node 6	$g_2^{(0)}+g_2^{(2)}$	$g_0^{(1)}+g_0^{(2)}$	$g_1^{(2)}$

Repair of stationary nodes

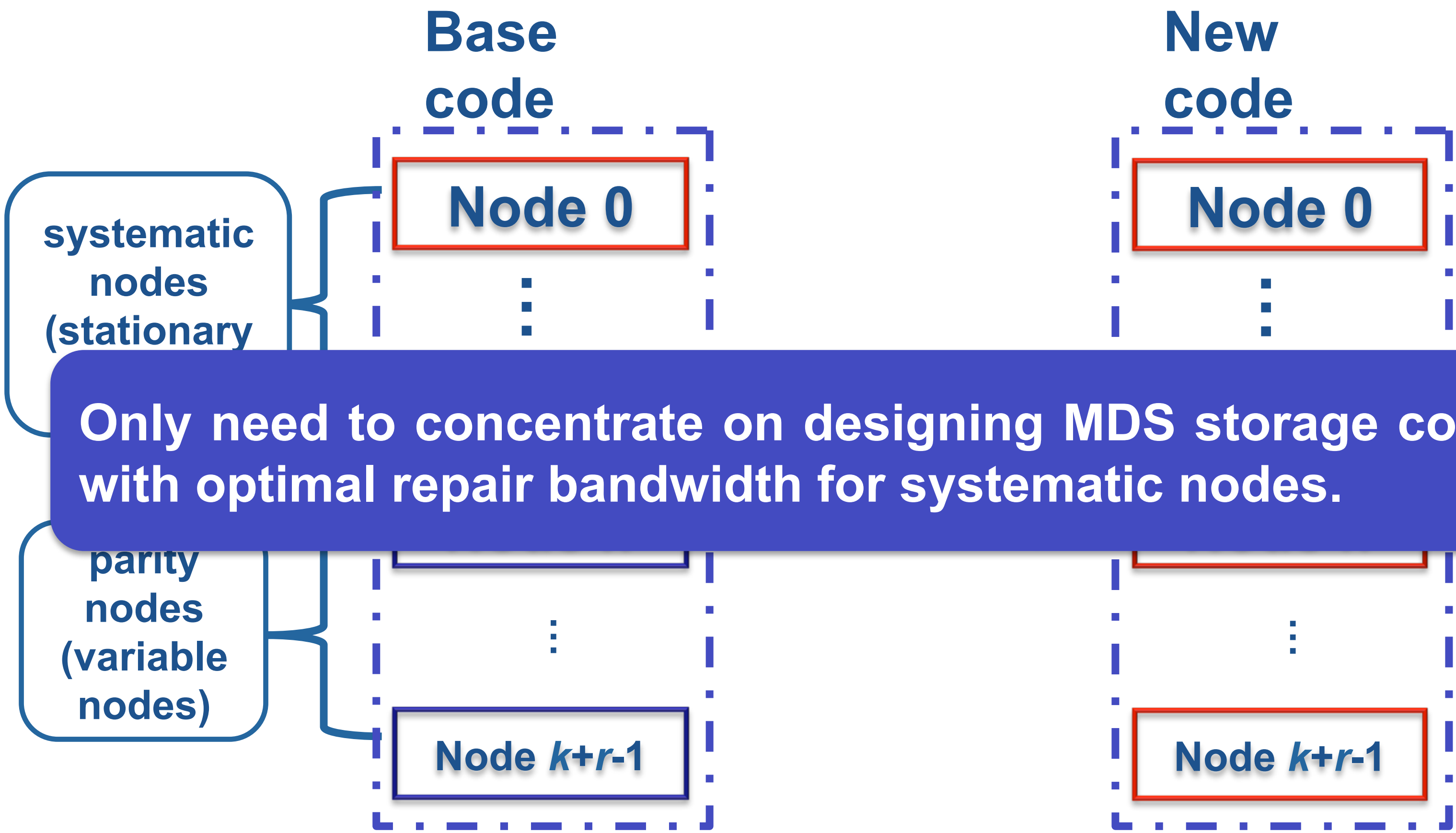
$$S_{0,1}f_1^{(i)}, S_{0,2}f_2^{(i)}, S_{0,3}f_3^{(i)}, S_{0,4}g_0^{(i)}, S_{0,5}g_1^{(i)}, S_{0,6}g_2^{(i)} \longrightarrow f_0^{(i)}$$

Download data $S_{i,j}f_j^{(l)}$

$$S_{i,k+j+l}(ag_{j+l}^{(l)} + g_{j+l}^{(j)})$$

	Instance 0	Instance 1	Instance 2
Node 0	$f_0^{(0)}$	$f_0^{(1)}$	$f_0^{(2)}$
Node 1	$S_{0,1}f_1^{(0)}$	$S_{0,1}f_1^{(1)}$	$S_{0,1}f_1^{(2)}$
Node 2	$S_{0,2}f_2^{(0)}$	$S_{0,2}f_2^{(1)}$	$S_{0,2}f_2^{(2)}$
Node 3	$S_{0,3}f_3^{(0)}$	$S_{0,3}f_3^{(1)}$	$S_{0,3}f_3^{(2)}$
Node 4	$S_{0,4}g_0^{(0)}$	$S_{0,5}(-g_1^{(1)}+g_1^{(0)})$	$S_{0,6}(-g_2^{(2)}+g_2^{(0)})$
Node 5	$S_{0,5}(g_1^{(0)}+g_1^{(1)})$	$S_{0,6}g_2^{(1)}$	$S_{0,4}(-g_0^{(2)}+g_0^{(1)})$
Node 6	$S_{0,6}(g_2^{(0)}+g_2^{(2)})$	$S_{0,4}(g_0^{(1)}+g_0^{(2)})$	$S_{0,5}g_1^{(2)}$

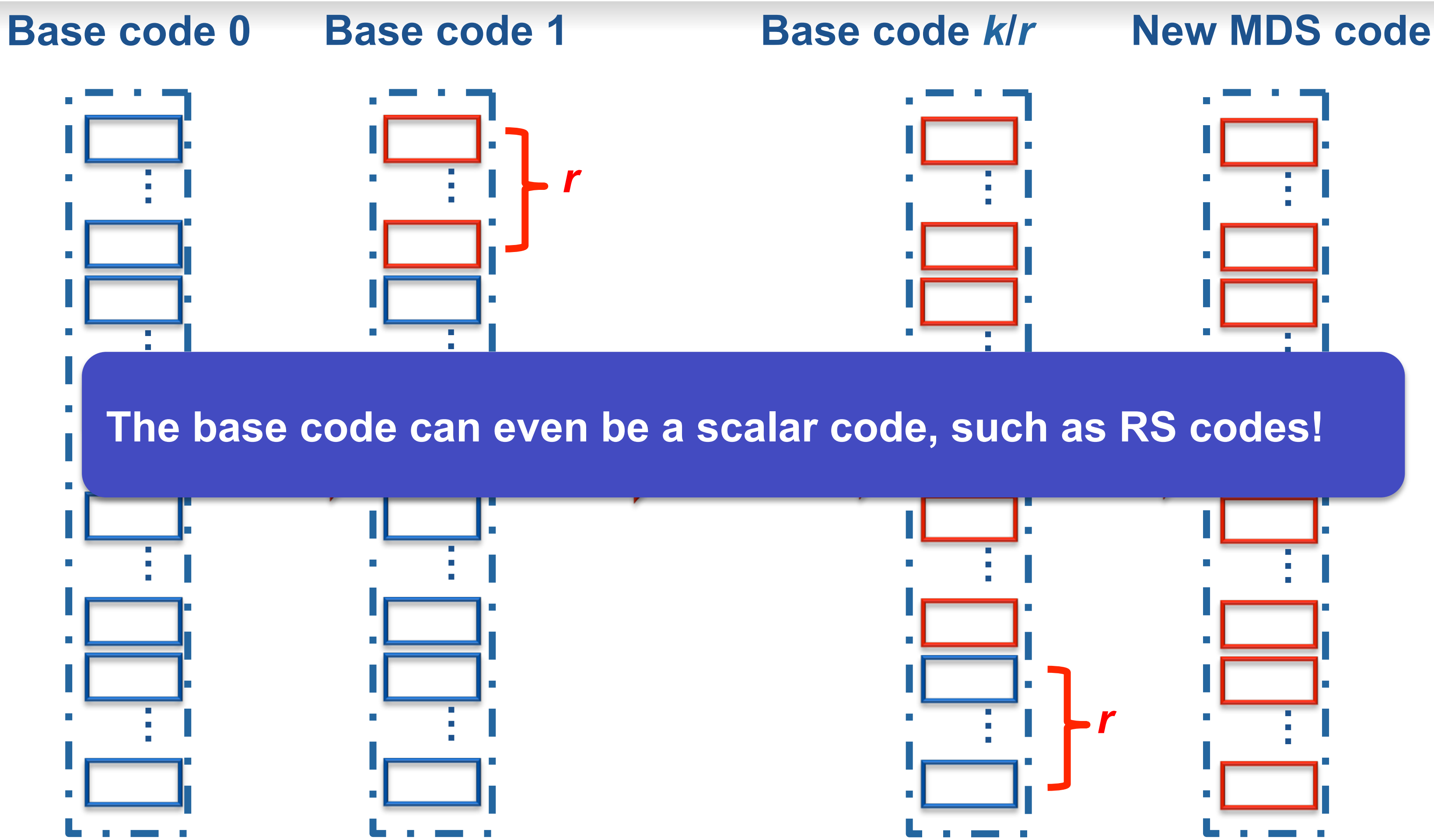
Application I



Only need to concentrate on designing MDS storage codes with optimal repair bandwidth for systematic nodes.

MDS storage code with optimal RB for systematic nodes → MDS storage code with optimal RB for all nodes

Application II



The base code can even be a scalar code, such as RS codes!

MDS code → MDS storage code with optimal repair bandwidth for all nodes

Remarks

MSR with optimal repair for all nodes

1.Li, Tang and Tian, Enabling All-Node-Repair in Minimum Storage Regenerating Codes, arXiv:1604.07671, April 2106.

2.Ye and Barg, Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization, arXiv:1605.08630, May 2016.

3.Sasidharan, Vajha, and Kumar, An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair, arXiv:1607.07335, July 2016.

MSR from MDS

1.Sasidharan, Vajha, and Kumar, An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair, arXiv:1607.07335, July 2016.

2.Li , Tang and Tian, A Generic Transformation for Optimal Repair Bandwidth and Rebuilding Access in MDS Codes," Proc. of the 2017 IEEE Internl. Symp. Inform. Th. , Aachen, Germany, June 2017.

A comparison with the recent results

A comparison of some key parameters between the (k+r, k) MSR codes

	Sub-packetization	Field size q	Systematic form
Ye-Barg code 1	r^{k+r}	$q > r(k+r)$	No
Hadmard design code employing our transformation	r^{k+1}	$q > rk$	Yes
Ye-Barg code 2	r^{k+r-1}	$q > k+r$	No
Zigzag code employing our transformation	r^{k}	$q = 3$ if $r = 2$ $q = 4$ if $r = 3$ $q > r^k \sum_{t=1}^{r} \binom{k-1}{t-1}\binom{r-1}{t-1}$ if $r > 3$	Yes
Ye-Barg code 3	$r^{\frac{k}{r}+1}$	$q > k+r$	No
Long MDS code employing our transformation	$r^{\frac{k}{r+1}+1}$	$q > r^{\frac{k}{r+1}+1} \sum_{t=1}^{r} \binom{k-1}{t-1}\binom{r-1}{t-1}$	Yes

Conclusions

- Proposed a framework of MDS storage code construction
  - with optimal repair property for systematic nodes
  - with optimal access property
  - with optimal update property

- Proposed a generic transformation of MDS storage code
  - from code with optimal repair property for systematic nodes to code with optimal repair property for all nodes
  - from scalar code to code with optimal repair property for all nodes


Thank You!