Codes for Big Data: Error-Correction for Distributed Storage

P. Vijay Kumar

Professor,
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore

European School of Information Theory (ESIT)
Chalmers University of Technology
Gothenburg

April 5, 2016

Acknowledgements

Research Collaborators

- Kannan Ramchandran
- Natalia Silberstein, Ankit S. Rawat, O. Ozan Koyluoglu, and Sriram Vishwanath,
- Chao Tian, Vaneet Aggarwal, Vinay A. Vaishampayan

- Srinivasan Narayanamurthy, Ranjit Kumar and Siddhartha Nandi (NetApp Inc., India)

Acknowledgements (continued)

Past and Present Students

- K. V. Rashmi and Nihar B. Shah (now pursuing PhD at UC Berkeley)
- Govinda M Kamath (now pursuing PhD at Stanford)

- Narayanamoorthy Prakash (currently a Post-Doc at MIT)
- Lalitha Vadlamani (currently faculty, IIITH)
- Birenjith Sasidharan (to graduate shortly)
- S. B. Balaji, Nikhil M. Krishnan, Myna Vajha (PhD students)

- Kaushik Senthoor, Gaurav Agrawal (now pursuing PhD at IITM)
- Kaushik Senthoor, Gaurav Agrawal (now pursuing PhD at UCLA)
- K. P. Prasanth (Master's student)

Thanks also to

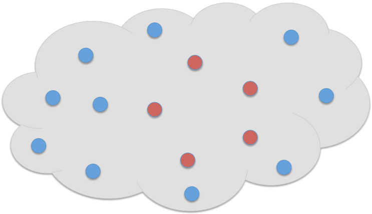
1. Fredrik Brännström,
2. Giuseppe Durisi, and
3. Alexandre Graell i Amat

for the invite and the super organization!

Organization

Module No	Topic
1	Distributed Storage, Reed-Solomon
2	Regenerating Codes
3	Interior Points, High-Rate Codes
4	Codes with Locality
5	Codes with Local Regeneration
6	Codes for Multiple Erasures
	List of References

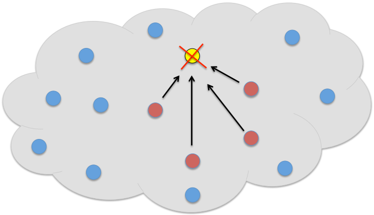
Distributed Storage Setting



- data pertaining to a single file is distributed across storage nodes

- nodes are inexpensive storage devices
  (a) prone to failure,
  (b) down for maintenance,
  (c) unavailable, busy serving other demands..

Distributed Storage Setting



- Need for efficient repair of a failed node arises
- Focus on
  (a) repair bandwidth - amount of data download
  (b) repair degree - number of helper nodes contacted

(the amount of data stored can be very very large $\Rightarrow$ "Big Data")

Just How Big is Big Data ?

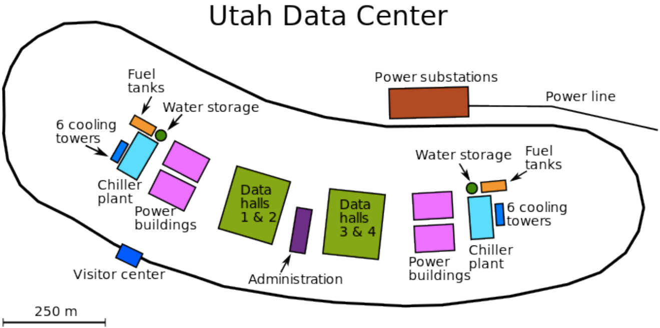
- Pictures from two different Data Centers..

A Recently Completed Large Data Center

Figure: The NSA Data Center in Utah.

- Estimated to store several between $3 - 12$ Exabytes!

  GigaByte $\rightarrow$ TeraByte $\rightarrow$ PentaByte $\rightarrow$ ExaByte $=$ One Billion GB!

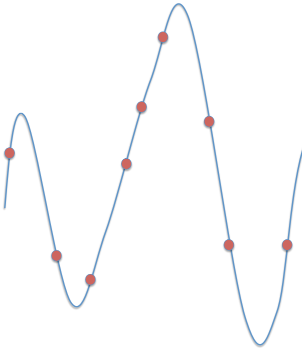


- Completed at an estimated cost of \$1.5 billion..
- Another \$2 billion for hardware, software, and maintenance
- 65 MW of power, costing about \$40 million per year
- use 1.7 million gallons of water per day

Reed-Solomon Codes

I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. J. SIAM, 1960.

The Underlying Principle of Reed-Solomon (RS) Codes



- Assume that this is the plot of a polynomial of degree 5
- then its values at any 6 of the 9 points shown are sufficient to determine its values everywhere else
- can use as an $[9, 6]$ erasure code (any 6 out of 9)

Example Finite Field $\mathbb{F}_8$ of size $2^3 = 8$

The field $\mathbb{F}_8$ consists of all polynomial expressions of the form

$$\sum_i a_i \alpha^i$$

involving an imaginary element α that satisfies the equation

$$\alpha^3 + \alpha + 1 \ = \ 0.$$

For this reason, we can write:

$$\mathbb{F}_8 \quad = \quad \{\sum_{i=0}^{2} a_i \alpha^i, \mid a_i \in \{0, 1\}\}.$$

Here, the coefficients $a_i \in \{0, 1\}$, commute multiplicatively with α^j, and arithmetic involving the a_i is carried out modulo 2:

$$\begin{aligned} a_i + a_j \quad &= a_i + a_j \pmod 2 \\ a_i a_j \quad &= a_i a_j \pmod 2. \end{aligned}$$

Conversion Table for Adding and Multiplying

Exponential rep.	Polynomial rep.
0	0
1	1
α	α
α^2	α^2
α^3	$\alpha + 1$
α^4	$\alpha^2 + \alpha$
α^5	$\alpha^2 + \alpha + 1$
α^6	$\alpha^2 + 1$
α^7	1

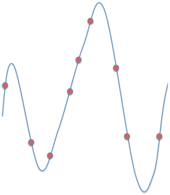
With this, we can add elements in the polynomial domain:

$$(\alpha^2 + \alpha) \; + \; (\alpha + 1) \;\; = \;\; \alpha^2 + 1$$

and use the exponential form to multiply:

$$\alpha^4\alpha^5 \; = \; \alpha^9 \; = \; \alpha^7\alpha^2 \; = \; \alpha^2.$$

Recovery by Solving a System of Linear Equations

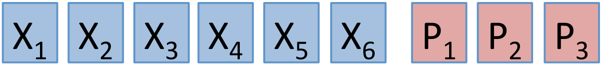


$$f(x) = \sum_{i=0}^{5} a_i x^i, \quad \text{(with } a_i \text{ lying in an appropriate finite field)}$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_6) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_1 & \cdots & x_1^5 \\ 1 & x_2 & \cdots & x_2^5 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_6 & \cdots & x_6^5 \end{bmatrix}}_{\substack{\text{Vandermonde matrix} \\ \text{(invertible)}}} \begin{bmatrix} a_0 \\ \\ \vdots \\ a_5 \end{bmatrix}$$

- The 6 coefficients $\{a_i\}_{i=0}^{5}$ can be recovered from *any* 6 values $\{f(x_i)\}_{i=1}^{6}$
- possesses the 'any-6-of-9' property

The Reed-Solomon Code in Operation



- the contents of a single data file split into 6 fragments and a Reed-Solomon code used to generate 3 additional redundant fragments which are stored in 9 nodes in the network
- each fragment represents a single symbol of the codeword

- the file can be recovered from any 6 fragments
- it can hence tolerate 3 node failures

- Overhead $= 50\%$ (sometimes, we will say overhead of 1.5)
- offers lower probability of data loss to triple replication (a competing code!), for lesser overhead

Maximum Distance Separable (MDS) Codes

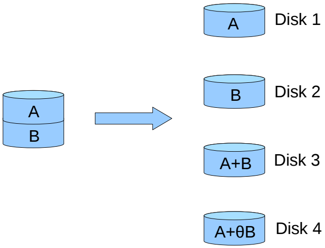
MDS codes are a class of codes that also possess the 'any k of n' property

1. this class includes Reed-Solomon codes
2. the minimum Hamming distance $d_{\min}$ between a pair of distinct codewords in an MDS code satisfies the Singleton bound

$$d_{\min} \quad \leq \quad n - k + 1,$$

with equality and the codes are hence said to be maximum distance separable.

An Example MDS Code Used in the Storage Industry



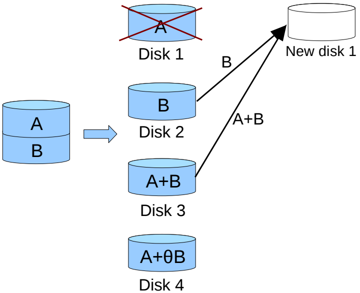
- $[4, 2]$ MDS code
- Can recover data by connecting to any 2 of 4 nodes
- In comparison with triple replication, offers robustness at smaller values of storage overhead

RAID: Redundant Array of Independent Disks

But How Well Does It Handle Node Failure ?

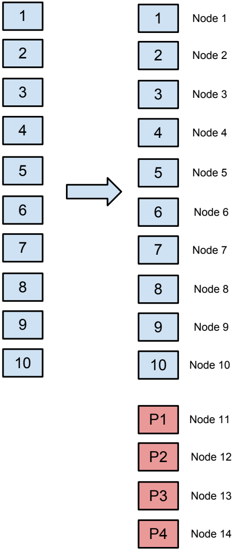
An obvious approach:

- Connect to any k nodes,
- Reconstruct entire data file,
- Reconstruct data stored in the node



But downloading 2 units of data to revive a node that stores 1 units of data is wasteful!

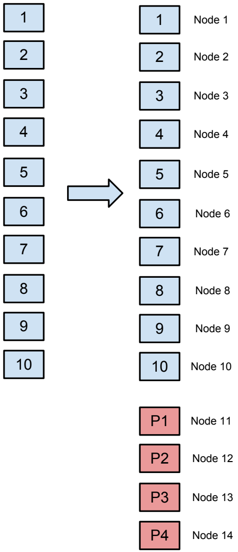
A Second Example: Facebook's HDFS-RAID Code



- $[14, 10]$ MDS code
- Can recover data by connecting to any 10 nodes
- Used in Facebook data centers
- HDFS $\equiv$ Hadoop Distributed File System

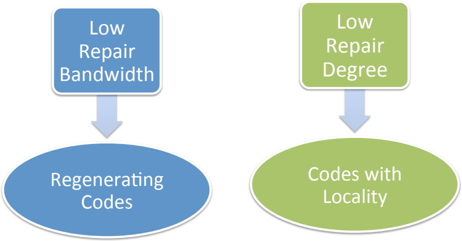
D. Borthakur, R. Schmit, R. Vadali, S. Chen, and P. Kling. "HDFS RAID." Tech talk. Yahoo Developer Network, Nov. 2010

How Well Does it Handle Node Failure ?



- Needs to connect to 10 nodes to repair a failed node
- This calls for interrupting operations in 10 nodes (apart from downloading the entire data file)
- 10 is the *repair degree*
- Are there better options ?

Two Problems – Two Solutions



(the focus of this tutorial is on the development of these two classes of codes)

- A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Trans. Inform. Th.*, Sep. 2010.
- P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the Locality of Codeword Symbols," *IEEE Trans. Inf. Theory*, Nov. 2012.

Push Back from Reed-Solomon Codes

1. Piggybacked RS codes
   - Improvements in repair of a modified RS code by repairing several codewords cooperatively
2. repairing RS codes using nonlinear operations

- K. V. Rashmi, N. B. Shah, and K. Ramchandran. A piggybacking design framework for read-and download-efficient distributed storage codes. In IEEE International Symposium on Information Theory, 2013.
- K. V. Rashmi, Nihar B. Shah, Dikang Gu, Hairong Kuang, Dhruba Borthakur, and Kannan Ramchandran , "A "Hitchhiker's" Guide to Fast and Efficient Data Reconstruction in Erasure-coded Data Centers, " ACM SIGCOMM, Aug 2014.
- Venkatesan Guruswami, Mary Wootters, "Repairing Reed-Solomon Codes," arXiv:1509.04764 [cs.IT] .

Piggy-Backing RS Codes - Encoding

a_1	a_2
b_1	b_2
$a_1 + b_1$	$a_2 + b_2$
$a_1 + 2b_1$	$a_2 + 2b_2$

$\Rightarrow$ (adding functions of col. 1 to entries in col. 2)

a_1	a_2
b_1	b_2
$a_1 + b_1$	$a_2 + b_2$
$a_1 + 2b_1$	$a_2 + 2b_2 + a_1$

$\Rightarrow$ (linear operations within the same node)

a_1	a_2
b_1	b_2
$a_1 + b_1$	$a_2 + b_2$
$a_1 + 2b_1 - (a_2 + 2b_2 + a_1)$	$a_2 + 2b_2 + a_1$

(each row is a node)

Piggy-Backing RS Codes - Repair

a_1	a_2
b_1	b_2
$a_1 + b_1$	$a_2 + b_2$
$2b_1 - a_2 - 2b_2$	$a_2 + 2b_2 + a_1$

$\Leftarrow$ The Code

$\cancel{a_1}$	$\cancel{a_2}$
b_1	b_2
$a_1 + b_1$	$a_2 + b_2$
$2b_1 - a_2 - 2b_2$	$a_2 + 2b_2 + a_1$

$\Leftarrow$ when node 1 fails

a_1	a_2
$\cancel{b_1}$	$\cancel{b_2}$
$a_1 + b_1$	$a_2 + b_2$
$2b_1 - a_2 - 2b_2$	$a_2 + 2b_2 + a_1$

$\Leftarrow$ when node 2 fails

(helper symbols in blue)

Efficient Repair of RS Codes

- Show that
  "$O(k)$ bits are necessary to recover a missing evaluation. In contrast, the traditional method of looking at k evaluations requires $\Omega(klog(k))$ bits. We also show that our result is optimal for linear methods, even up to the leading constants."

- Venkatesan Guruswami and Mary Wootters, "Repairing Reed-Solomon Codes," arXiv:1509.04764v1 [cs.IT] for this version.

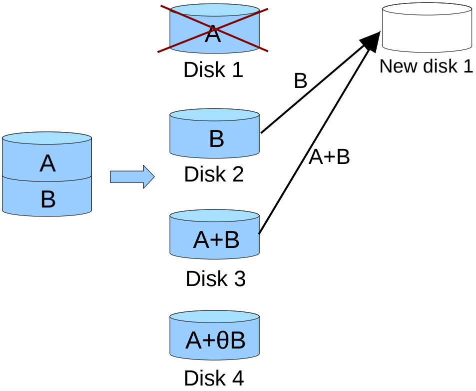
Regenerating Codes

Dimakis, Godfrey, Wu,Wainwright, Ramchandran, T-IT, Sep. 2010, Communications Society & Information Theory Society Joint Paper Award.

RAID Codes not very Efficient at Handling Node Repair

Approach to node repair:

- Connect to any k nodes,
- Reconstruct entire data file,
- Reconstruct data stored in the node

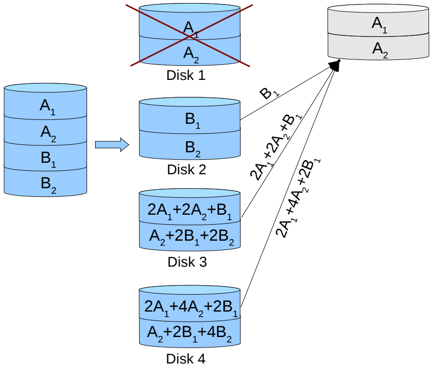


But downloading 2 units of data to revive a node that stores 1 unit of data is wasteful!

(focus here is on minimizing repair bandwidth)

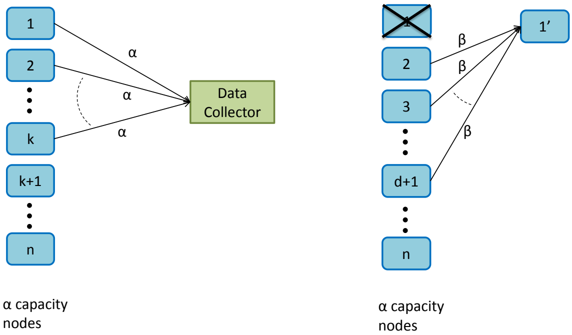
An Improved (Regenerating) Code

- Here, each node now stores two "half-symbols"
- We download 3 half-symbols as opposed to 2 full-symbols
  - vector symbol alphabet $\Rightarrow$ $\mathbb{F}_q^2$ versus $\mathbb{F}_{q^2}$



Regenerating Codes - Formal Definition

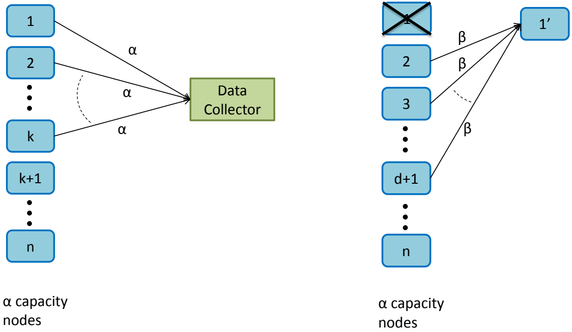
Parameters: $(\ (n, k, d),\ (\alpha, \beta),\ B,\ \mathbb{F}_q\ )$



- Data to be recovered by connecting to any k of n nodes
- Nodes to be repaired by connecting to any d nodes, downloading β symbols from each node; ($d\beta <<$ file size B)
- Differentiate between functional and exact repair

Regenerating Codes - Formal Definition

Parameters: $(\ (n, k, d),\ (\alpha, \beta),\ B,\ \mathbb{F}_q\ )$

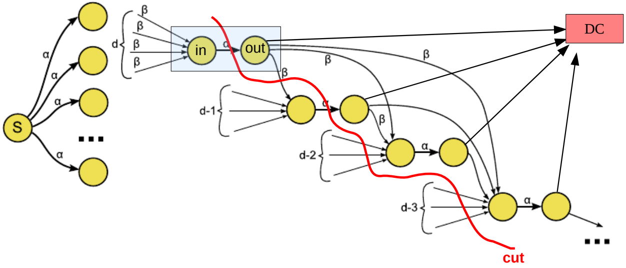


- Data to be recovered by connecting to any k of n nodes
- Nodes to be repaired by connecting to any d nodes, downloading β symbols from each node; ($d\beta <<$ file size B)
- Differentiate between functional and exact repair

Cut-Set Bound from Network Coding

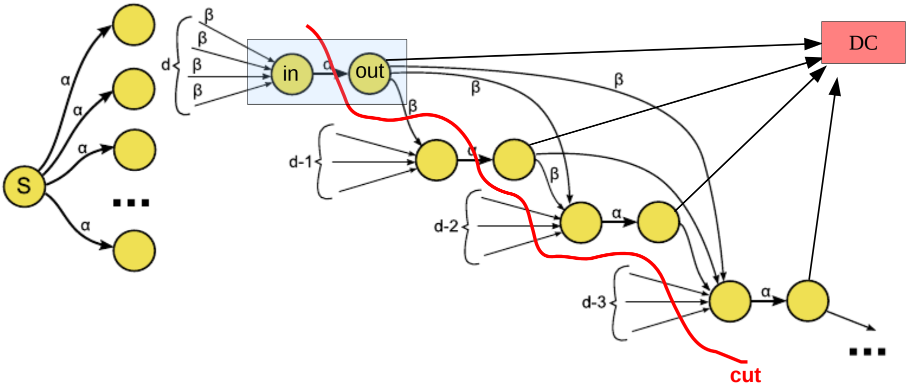
Given code parameters $\{[n, k, d], (\alpha, \beta)\}$:

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



(can be shown to be achievable under functional repair)

Dimakis, Godfrey, Wu,Wainwright, Ramchandran, T-IT, Sep. 2010
Wu, IEEE JSAC, Feb. 2010.



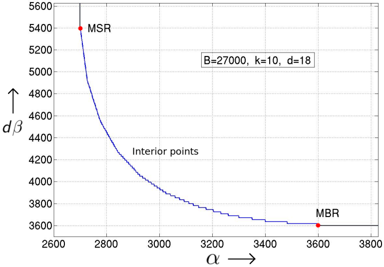
(the capacity of the cut shown equals $\alpha + \alpha + (d-2)\beta + (d-3)\beta$)

The Storage-Repair Bandwidth Tradeoff

The upper bound on file size:

$$B \leq \sum_{i=1}^{k} \min\{\alpha, (d-i+1)\beta\} \quad \text{(multiple } (\alpha, \beta) \text{ pairs can achieve bound)}$$

- Tradeoff curve drawn for fixed (k, d), B.
- Extreme points: MSR & MBR
  - MSR=Minimum Storage Regenerating $\alpha = (d-k+1)\beta$
  - MBR=Minimum Bandwidth Regenerating $\alpha = d\beta$



$(k, d) = (120, 129)$, $B = 725360$

File Sizes

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

1. MSR Code:

$$B = \alpha k$$

   - Hence $q^B = q^{\alpha k} = (q^\alpha)^k = (q^\alpha)^{n-d_{\min}+1}$ achieves the Singleton bound on code size over an alphabet $\mathbb{F}_q^\alpha$ of size q^α.
   - Hence MSR codes are MDS!

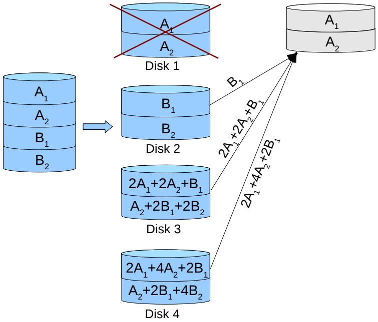
2. MBR File size:

$$B = \sum_{i=1}^{k} (d-i+1)\beta = \left(dk - \binom{k}{2}\right)\beta.$$

AN EXAMPLE MSR CODE

The (Previously Seen) Example MSR Code

- Parameters: $\{(n = 4, k = 2, d = 3),\ (\alpha = 2, \beta = 1),\ B = 4\}$
- A vector MDS code
- $\alpha = (d - k + 1)$ (minimum possible) and $B = \alpha k$



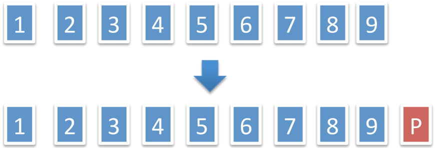
At the other end of the tradeoff,

AN EXAMPLE MBR CODE

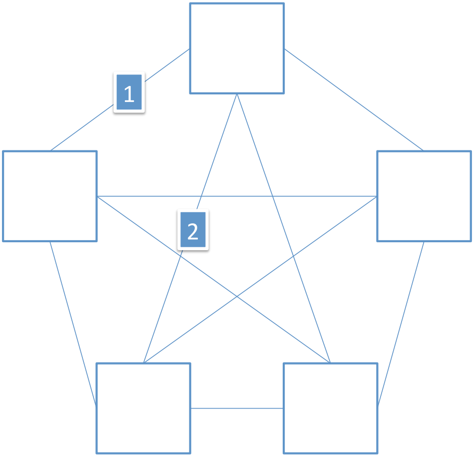
(aka “The Repair-by-Transfer” MBR Code)

Shah, Rashmi, PVK, Ramchandran, T-IT, Mar. 2012.

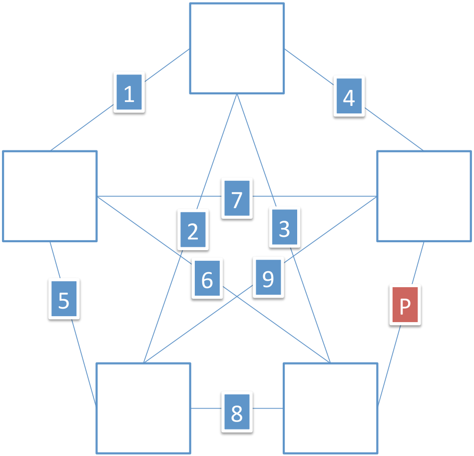
Step 1: Add an Extra Parity to the 9 Units of Data



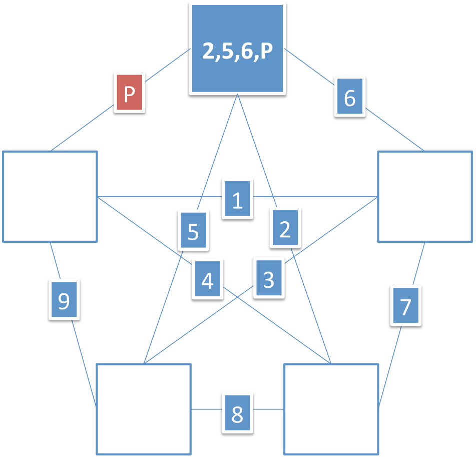
Step 2: Set up Completely-Connected Pentagon (10 Edges)



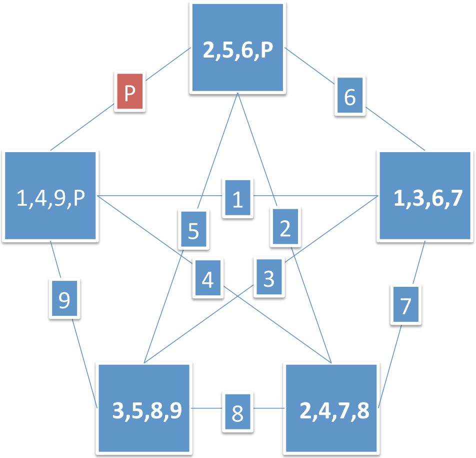
Step 3: Place Coded Data on Edges



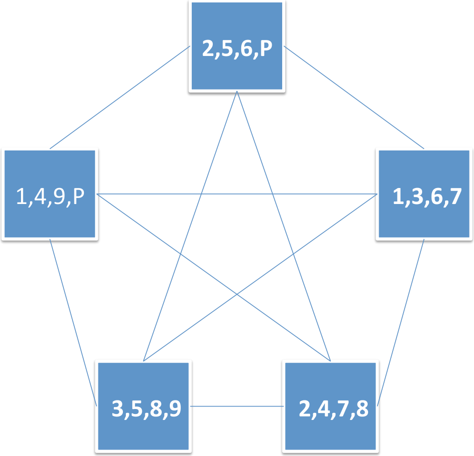
Step 4: Load Data from Edges onto Nodes



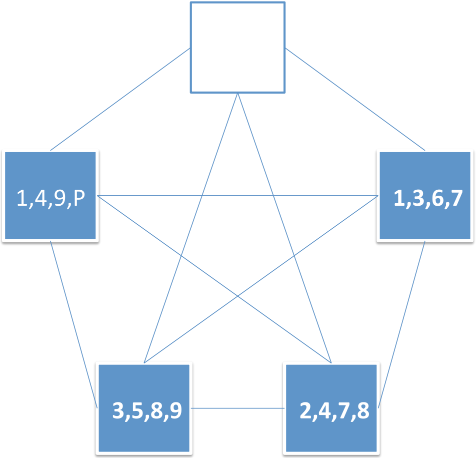
Step 4: Transfer Data from Edges into Nodes



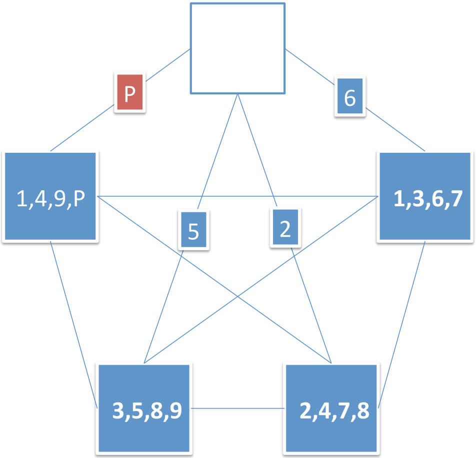
End of Encoding Procedure



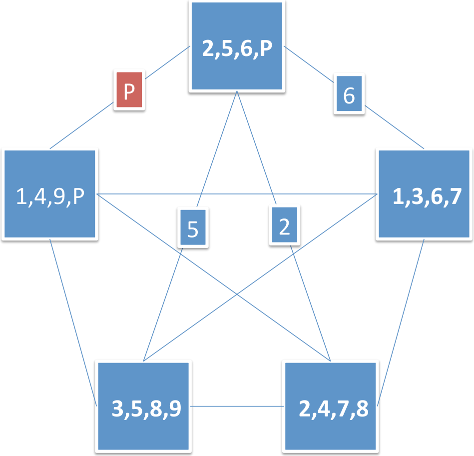
Node Failure



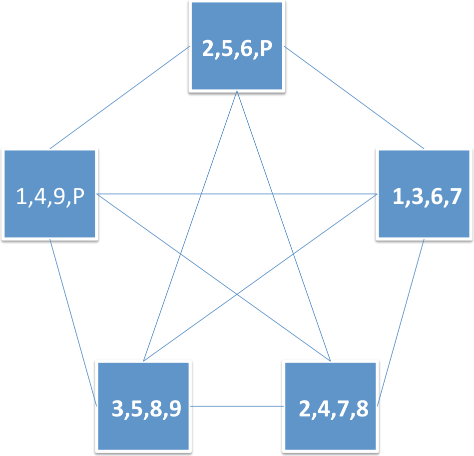
Node Repair



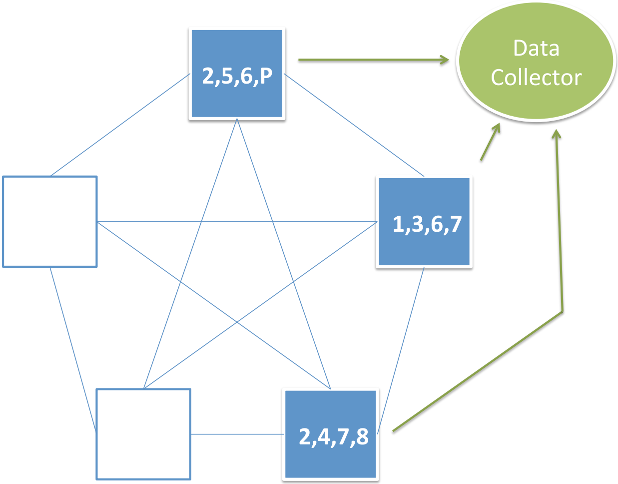
Node Repair



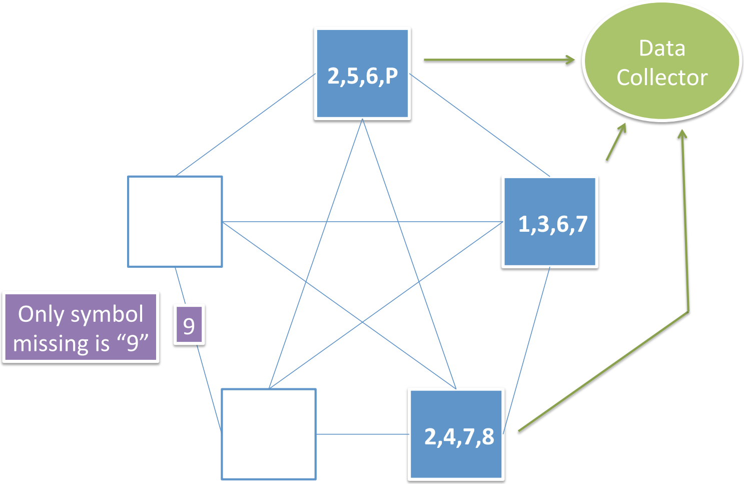
Node Repair Complete



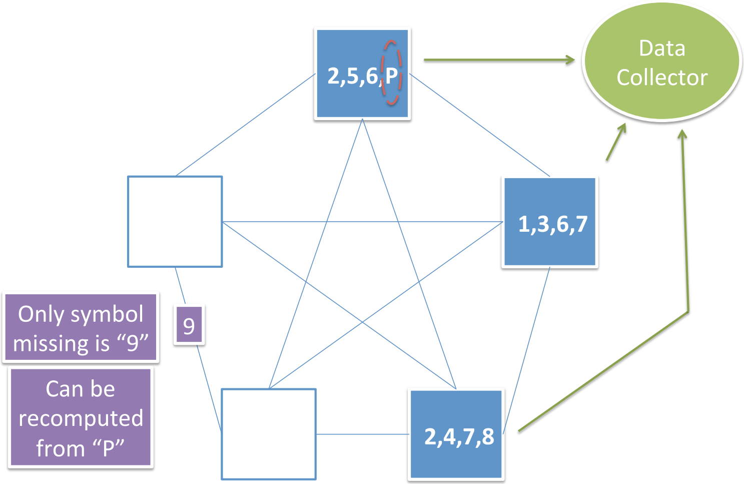
Data Collection



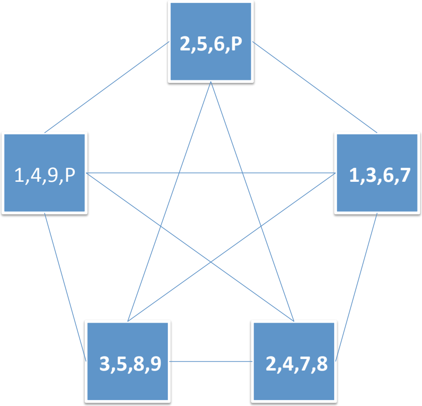
Data Collection



Data Collection Complete



Pentagon Code Node Downloads only as Much as it Stores



(hence, is repair-bandwidth efficient)

THE PRODUCT MATRIX CODE

Rashmi, Shah and PVK, T-IT, AUG. 2011, 2011-12 IEEE Data Storage Best Paper and Best Student Paper Award.

Product-Matrix Framework

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

- M : Message matrix
  - Contains message symbols with some message symbols repeated
  - Possesses a block-symmetry property

- Ψ : Encoding matrix
  - Used to disperse information across the nodes
  - Independent of message symbols

- C : Code matrix
  - Each row represents one node
  - i^{th} node stores: $\underline{\psi}_i^t M$

The Product-Matrix MBR (PM-MBR) Code

- $\alpha = d$
- $B = kd - \binom{k}{2} \quad \rightarrow \quad B = \binom{k+1}{2} + k(d-k)$

- Let S be a $(k \times k)$ symmetric matrix with $\binom{k+1}{2}$ distinct message symbols
- Let T be a $(k \times (d-k))$ matrix with $k(d-k)$ distinct message symbols
- thus all message symbols are accounted for

Product-matrix MBR Code

- Message matrix $$\underbrace{M}_{d\times d} = \begin{bmatrix} \underbrace{S}_{k\times k} & \underbrace{T}_{k\times(d-k)} \\ \underbrace{T^t}_{(d-k)\times k} & \underbrace{0}_{k\times(d-k)} \end{bmatrix} \quad \text{(symmetric)}$$

- Encoding matrix $$\underbrace{\Psi}_{n\times d} = \begin{bmatrix} \underbrace{\Phi}_{n\times k} & \underbrace{\Delta}_{n\times(d-k)} \end{bmatrix}$$

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

  e.g., Cauchy, Vandermonde matrix

Product-matrix MBR Code : Data Reconstruction

Node i passes: $\underline{\psi}_i^t M$

aggregate $\downarrow$

$\Psi_{\text{DC}} M$

$(\Psi_{\text{DC}} = [\Phi_{\text{DC}} \quad \Delta_{\text{DC}}]$ is $(k \times d))$

$\downarrow$

decode

$$\left[\begin{array}{cc} \Phi_{\text{DC}} S + \Delta_{\text{DC}} T^t & \Phi_{\text{DC}} T \end{array}\right]$$

$\downarrow$

Φ_{DC} is $k \times k$, invertible

Decode T

$\downarrow$

Subtract $\Delta_{\text{DC}} T^t$, Decode S

$$M = \left[\begin{array}{cc} S & T \\ T^t & 0 \end{array}\right]$$

$$\Psi = \left[\begin{array}{cc} \Phi & \Delta \end{array}\right]$$

$$C = \Psi M$$

Product-matrix MBR Code : Exact Regeneration

Replacement node f needs: $\underline{\psi}_f^t M$

Helper node i, $1 \leq i \leq d$ stores: $\underline{\psi}_i^t M$

Helper node i passes: $\underline{\psi}_i^t M \underline{\psi}_f$

aggregate $\downarrow$

$\Psi_{\text{repair}} M \underline{\psi}_f$

(Ψ_{repair} is $d \times d$, invertible)

$\downarrow$

$M \underline{\psi}_f$

(M is symmetric)

$\downarrow$

$\underline{\psi}_f^t M$

$$M = \begin{bmatrix} S & T \\ T^t & 0 \end{bmatrix}$$

$$\Psi = \begin{bmatrix} \Phi & \Delta \end{bmatrix}$$

$$C = \Psi M$$

The Product-matrix MSR Code Parameters

- Here again $\beta = 1$
- $\alpha = d - k + 1$

The MSR point-Numerology

- $d < 2k - 3$ not possible with $\beta = 1$
- This code is designed for $d \geq 2k - 2$
- Choose $d = 2k - 2$ first, then extend to higher d

- Gives

$$\begin{aligned} k &= \alpha + 1 \\ d &= 2\alpha \\ B &= \alpha(\alpha + 1) \end{aligned}$$

- S_1, S_2: $(\alpha \times \alpha)$ symmetric matrices with $\frac{\alpha(\alpha+1)}{2}$ distinct message symbols each

The Product-Matrix MSR Code

- Message matrix $$\underbrace{M}_{d\times\alpha} = \begin{bmatrix} \underbrace{S_1}_{\alpha\times\alpha} \\ \underbrace{S_2}_{\alpha\times\alpha} \end{bmatrix}$$

- Encoding matrix $$\underbrace{\Psi}_{n\times d} = \begin{bmatrix} \underbrace{\Phi}_{n\times\alpha} & \underbrace{\Lambda\Phi}_{n\times\alpha} \end{bmatrix}$$

  Φ: any α rows linearly independent
  Λ: $n \times n$ diagonal matrix with the diagonal elements distinct
  Ψ: any d rows linearly independent
  e.g., Vandermonde

The Product-Matrix MSR Code-Data Reconstruction

$$\text{Node } i \text{ passes: } \underline{\psi}_i^t M$$

$$\text{aggregate}\downarrow$$

$$\Psi_{\text{DC}} M$$

$(\Psi_{\text{DC}} = [\Phi_{\text{DC}} \quad \Lambda_{\text{DC}}\Phi_{\text{DC}}]$ is $k \times d)$

$$\downarrow$$

$$[\Phi_{\text{DC}} S_1 + \Lambda_{\text{DC}}\Phi_{\text{DC}} S_2]$$

$$\downarrow$$

$$[\Phi_{\text{DC}} S_1 \Phi_{\text{DC}}^t + \Lambda_{\text{DC}}\Phi_{\text{DC}} S_2 \Phi_{\text{DC}}^t]$$

$$\downarrow$$

$$[P + \Lambda_{\text{DC}} Q]$$

(P and Q symmetric)

$$\downarrow$$

$$(i,\ j) : P_{ij} + \lambda_i Q_{ij},\ \ (j,\ i) : P_{ij} + \lambda j Q_{ij}$$

(Solve for P and Q)

$$\downarrow$$

Recover S_1 and S_2

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

$$\Psi = \begin{bmatrix} \Phi & \Lambda\Phi \end{bmatrix}$$

$$C = \Psi M$$

The Product-Matrix MSR Code-Exact Regeneration

Replacement node f needs: $\underline{\psi}_f^t M$

Helper node i stores: $\underline{\psi}_i^t M$

Helper node i passes: $\underline{\psi}_i^t M \underline{\phi}_f$

aggregate $\downarrow$

$$\Psi_{\text{rep}} M \underline{\phi}_f$$

(Ψ_{rep} is $d \times d$, invertible)

$$\downarrow$$

$$M\underline{\phi}_f = \begin{bmatrix} S_1\underline{\phi}_f \\ S_2\underline{\phi}_f \end{bmatrix}$$

$$\downarrow$$

$$\underline{\phi}_f^t S_1 + \lambda_f \underline{\phi}_f^t S_2 \;=\; \underline{\psi}_f^t M$$

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

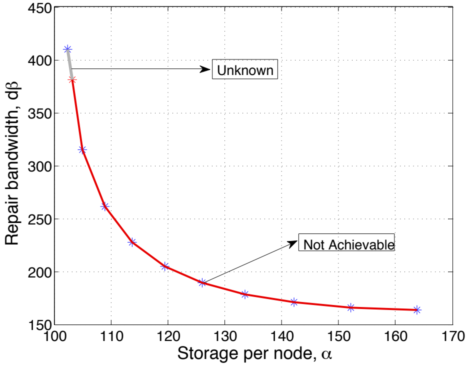
$$\Psi = \begin{bmatrix} \Phi & \Lambda\Phi \end{bmatrix}$$

$$C = \Psi M$$

INTERIOR POINTS OF THE TRADEOFF

Interior Points Not-Achievable Under Exact Repair!

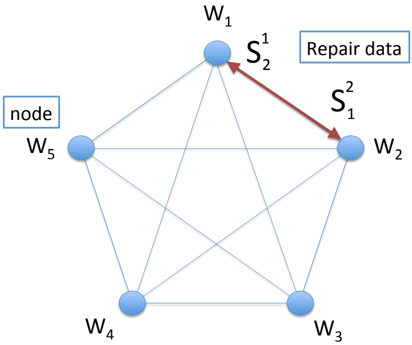
No exact-repair code can achieve an interior point on the tradeoff...



(However, can exact-repair codes approach the tradeoff asymptotically, i.e., as $B \rightarrow \infty$?)

(Shah, Rashmi, PVK, Ramchandran, T-IT, Mar. 2012),

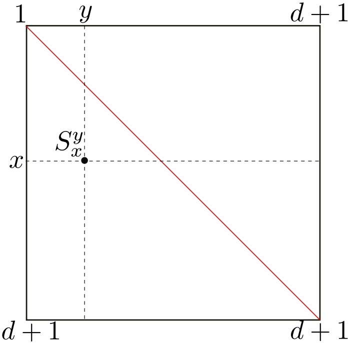
Explaining Why Not Achievable - Notation



- n nodes
- ith node stores α symbols, random variable W_i
- $S_x^y \Rightarrow$ the β symbols sent from x to repair node y

The Repair Matrix $\mathcal{R}$

- S_x^y is the repair data sent from node x to node y



$((d+1) \times (d+1))$

More Notation

n random variables:

$$\{W_i \mid 1 \le i \le n\}.$$

A further $n(n-1)$ random variables:

$$\{S_i^j \mid 1 \le i, j \le n, \quad i \ne j\}.$$

Hence, in all, n^2 random variables.

Let $\mathcal{B}$ denote the data file and

$$\mid \mathcal{B} \mid = B.$$

Constraints

$H(W_i) \leq \alpha,$	entropy of ith node
$H(S_i^j) \leq \beta,$	entropy of repair data

$H(\mathcal{B} \mid W_A) = 0, \mid A \mid = k,$	data collection property
$H(W_i \mid \mathcal{B}) = 0,$	node contents a function of file data
$H(S_i^j \mid W_i) \; = \; 0,$	repair data draws from node contents
$H(W_i \mid S_A^i) \; = \; 0,$ if $\mid A \mid = d$ and $i \notin A$	repair property

Some Background: Non-Existence Proof (Exact-Repair)

1. MSR $\Leftrightarrow\ \alpha = (d-k+1)\beta$, MBR $\Leftrightarrow\ \alpha = d\beta$
2. Interior: $\Leftrightarrow\ \ \alpha = (d-\mu)\beta$, $1 \leq \mu \leq (k-2)$

We assume wolog $(n = d+1)$, as restriction to $(d+1)$ nodes is also a regenerating code:

$$\text{Parameters: } (\ (n' = d+1, k, d),\ (\alpha, \beta),\ B,\ \mathbb{F}_q\ )$$

Exact-Repair File Size Bound

Let

$$\begin{aligned} [d+1] &= X \,\dot{\cup}\, Y \,\dot{\cup}\, Z \\ \mid X \mid &= \mu + 1 \\ \mid Y \mid &= k - (\mu + 1) \\ \mid Z \mid &= (d + 1 - k) \end{aligned}$$

Then

$$\begin{aligned} B &= H(W_X, S_Y, S_Z^Y) \\ S_Y &= \{S_i^j \mid i, j \in Y, i > j\} \\ S_Z^Y &= \{S_z^y \mid z \in Z, y \in Y\} \end{aligned}$$

Exact-Repair File Size Bound

Turns out that if an exact-repair code meets the cut-set bound, in the inequalities

$$
\begin{aligned}
B &= H(W_X, S_Y, S_Z^Y) \\
&= H(W_X) + H(S_Y \mid W_X) + H(S_Z^Y \mid W_X, S_Y) \\
&\leq H(W_X) + H(S_Y) + H(S_Z^Y) \\
&\leq \mid X \mid \alpha + \mid S_Y \mid \beta + \mid S_Z^Y \mid \beta,
\end{aligned}
$$

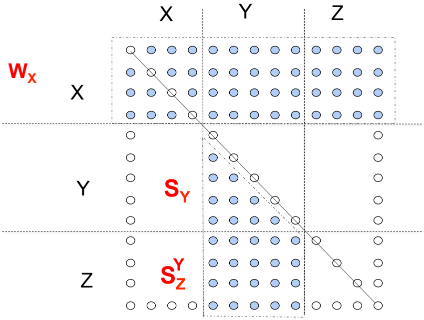
we must have, equality throughout, i.e.,

$$B = \mid X \mid \alpha + \mid S_Y \mid \beta + \mid S_Z^Y \mid \beta.$$

Non-Existence via Properties of the Repair Matrix $\mathcal{R}$

Assuming the existence of an optimal exact-repair code, we must have:

$$B = H(W_X, S_Y, S_Z^Y) \;\; = \;\; |\, X \,|\, \alpha + \;|\, S_Y \,|\, \beta + \;|\, S_Z^Y \,|\, \beta.$$

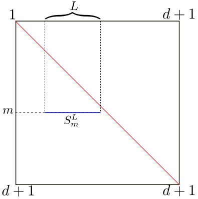


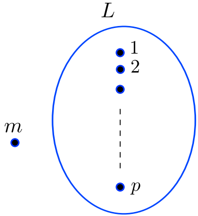
- Turns out however, every row of $\mathcal{R}$ has entropy at most β - contradiction!

Shah, Rashmi, PVK, Ramchandran, T-IT, 2012.

Explaining Why Rows Have Small Entropy

Goal: Explain why every row of $\mathcal{R}$ has entropy at most β. In figure below, $|L| = p = (\mu + 1)$.





$$H(S_m^L) \;=\; \underbrace{H(S_m^L \mid W_L)}_{\leq (\mu+1) H(S_m^{\ell_0} | W_L) = 0} + \underbrace{I(S_m^L : W_L)}_{\leq I(W_m; W_L) \leq \beta} \;\leq \beta.$$

Because $\mid L \mid = (\mu + 1)$ is

- large enough to permit interference cancellation to take place while passing repair information
- small enough that the mutual information is limited by β

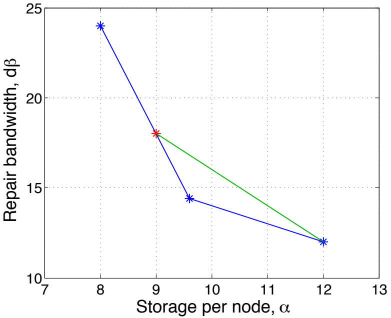
The Computation

We have:

$$
\begin{aligned}
H(S_m^L) &= H(S_m^L \mid W_L) + I(S_m^L : W_L) \\
&\le \ell \left\{ H(W_L/S_m^{\ell_0}) + H(S_m^{\ell_0}) - H(W_L) \right\} \\
&\quad + \{H(W_m) + H(W_L) - H(W_L, W_m)\} \\
&\le \ell \underbrace{\{\mu\alpha + (\alpha - \beta) + \beta - (\mu+1)\alpha\}}_{=0} \\
&\quad + \underbrace{\{(\mu+1)\alpha + \alpha - (\mu+1)\alpha + (\alpha - \beta)\}}_{=\beta} \\
&= \beta.
\end{aligned}
$$

CAN AN INTERIOR POINT BE APPROACHED ?

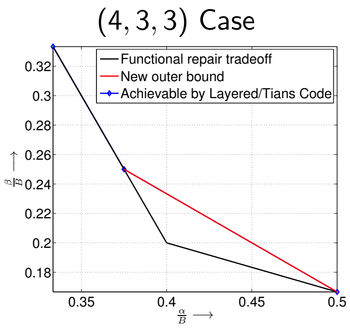
No! From Characterization of the $(4, 3, 3)$ Tradeoff

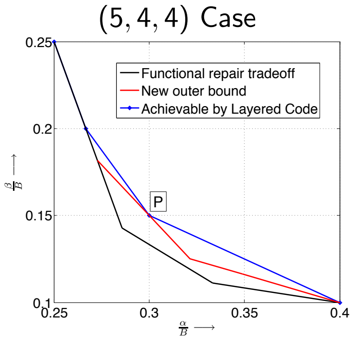


- FR Tradeoff = Blue
- ER Tradeoff = Max{Blue, Green}
- Chao Tian provided an explicit proof by using Raymond Yeung's ITIP framework to extract an additional inequality for the $(4, 3, 3)$ case.

A Dozen Bottles of Ouzo!

Our Subsequent Results (2014)





- First outer bound on the ER tradeoff that improves upon the FR tradeoff for all $[n, k, d]$
- Coincides with the ER tradeoff characterized by Tian for the $[4, 3, 3]$ case
- Shown alongside is the outer bound in the $[5, 4, 4]$ case
- In the $[5, 4, 4]$ case, bound coincides at one point P with performance of a layered code.
- First instance of an optimal code operating off of the FR tradeoff.

Layered code: (Tian, Sasidharan, Aggarwal, Vaishampayan, PVK, T-IT, Apr 2015)

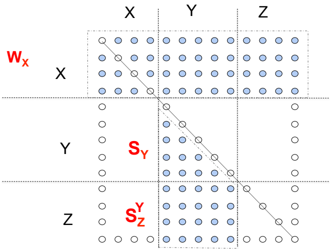
Our Approach

Let $\mathcal{T}$ denote the 'trapezium-shaped' region of the repair matrix:

$$\mathcal{T} \;=\; S_Y \,\dot{\cup}\, S_Z^Y \;\subseteq\; \mathcal{R}$$

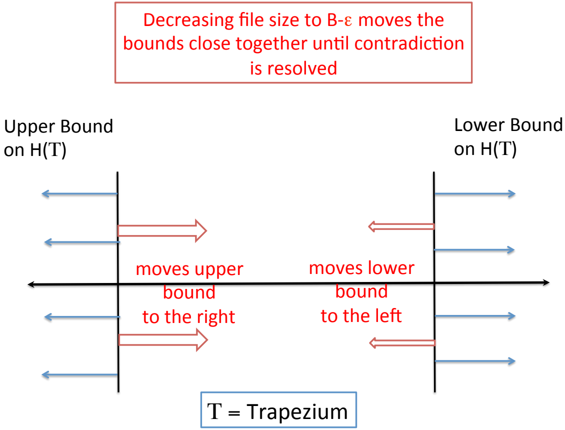
Assuming the existence of an optimal exact-repair code, we must have:

$$H(\mathcal{T}) \;=\; |\,\mathcal{T}\,|\,\beta$$



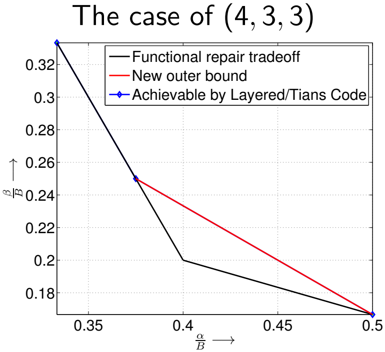
- On the other hand, every row of $\mathcal{T}$ has entropy at most β, this is a large gap which we exploit!

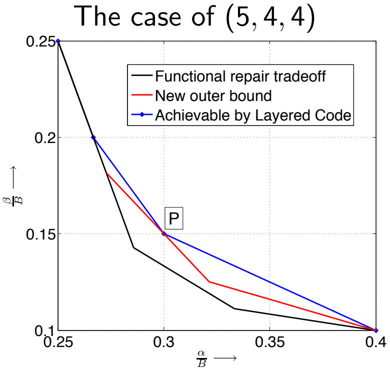
Approach to Deriving the New Bound



This leads to an new tradeoff as shown earlier.

The New Outer Bound





- Provides a new outer bound on ER tradeoff for all $[n, k, d]$
- Bound coincides with the tradeoff characterized by Tian in $[4, 3, 3]$ case.
- The bound in $[5, 4, 4]$ case coincides at one point P with an achievable region by layered codes.
- First instance of an optimal code operating off of the FR tradeoff.

Subsequent Work

- Iwan Duursma, "Outer bounds for exact repair codes," 2014.
- Iwan Duursma, "Shortened regenerating codes," 2015.
- Soheil Mohajer & Ravi Tandon. Exact Repair for Distributed Storage Systems: Partial Characterization via New Bounds, 2015
- Chao Tian, A Note on the Rate Region of Exact-Repair Regenerating Codes, 2015
- N. Prakash, M. Nikhil Krishnan, "The Storage-Repair-Bandwidth Trade-off of Exact Repair Linear Regenerating Codes for the Case $d = k = (n - 1)$", 2015.

(not described here for lack of time)

CONSTRUCTION OF HIGH-RATE MSR CODES

Constructions of MSR Codes (Rate $R \leq \frac{1}{2}$)

1. K. V. Rashmi, Nihar B. Shah and PVK, "Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction," IT-Trans, August 2011.

2. Changho Suh and Kannan Ramchandran, "Exact-Repair MDS Code Construction Using Interference Alignment," IT-Trans, March 2011.
   - Nihar Shah, K. V. Rashmi, PVK and Kannan Ramchandran, "Interference Alignment in Regenerating Codes for Distributed Storage: Necessity and Code Constructions," IT-Trans, April 2012.

Constructions of High-Rate MSR Codes (Rate $R > \frac{1}{2}$)

1. Viveck R. Cadambe, SyedAli Jafar, Hamed Maleki, Kannan Ramchandran and Changho Suh, "Asymptotic Interference Alignment for Optimal Repair of MDS Codes in Distributed Storage," IT-Trans, May 2013. (establish existence)
2. D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair Optimal Erasure Codes through Hadamard Designs," IT-Trans, May 2013. (construction for 2 parities)
3. Itzhak Tamo, Zhiying Wang, and Jehoshua Bruck, "Zigzag Codes: MDS Array Codes With Optimal Rebuilding," IT-Trans, March 2013. (repair systematic nodes)
4. Z. Wang, I. Tamo, J. Bruck, "On Codes for Optimal Rebuilding Access," *Allerton*, 2011 (also repair parity)

Sub-Packetization Level

1. Bound in [1]

$$\log_2(\alpha)\left(\log_\delta(\alpha)+1\right) \geq \frac{k-1}{2}$$
$$\delta = 1+\frac{1}{r-1}, \quad r=(n-k).$$

2. Construction in [2]

$$\alpha = r^{k+1}$$

3. Present Construction

$$\alpha = r^{\frac{n}{r}}$$

[1] Sreechakra Goparaju, Itzhak Tamo, and Robert Calderbank, "An Improved Sub-Packetization Bound for Minimum Storage Regenerating Codes," IT-Trans, May 2014.

[2] Z. Wang, I. Tamo, J. Bruck, "On Codes for Optimal Rebuilding Access," *Allerton*, 2011.

Sub-Packetization Level

- Present Construction

$$
\begin{aligned}
\alpha &= r^{\frac{n}{r}} \\
r &= (n-k)
\end{aligned}
$$

Parameter t	Rate $R = \frac{t-1}{t}$	Sub-packetization level α
$t = 3$	$\frac{2}{3}$	r^3
$t = 4$	$\frac{3}{4}$	r^4
$t = 5$	$\frac{4}{5}$	r^5

Construction Builds on the Earlier Work ...

- Itzhak Tamo, Zhiying Wang, and Jehoshua Bruck, "Zigzag Codes: MDS Array Codes With Optimal Rebuilding," IT-Trans, March 2013
- Z. Wang, I. Tamo, J. Bruck, "On Codes for Optimal Rebuilding Access," *Allerton*, 2011

How We Will Explain Construction ...

- Parity-Check Point of View
- First present a simplistic view of parities that will repair but cannot handle data collection
- Will then refine this
- Will then refine this further (this will now permit data collection as desired)

Parameters of Construction

$$\boxed{\text{Parameters: } (\ [n=6, k=4, d=5],\ [\alpha, \beta],\ B,\ \mathbb{F}_q\ )}$$

General	General	in Example
n	tq	6
k	$(t-1)q$	4
d	$(n-1)$	5
α	q^t	8
β	q^{t-1}	4
r	q	2
Rate	$\frac{t-1}{t}$	$\frac{2}{3}$
α	$r^{\frac{n}{r}}$	$2^{\frac{6}{3}} = 8$

Notation Used in Construction

$$\boxed{\text{Parameters: } (\ [n,k,d],\ [\alpha,\beta],\ B,\ \mathbb{F}_q\ )}$$

	Node 1	Node 2	$\cdots$	Node n
First symbol in node				
Second symbol in node				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
Last αth symbol in node				

$(n \times \alpha)$ codeword array

$$\text{Code symbol } C(\underbrace{\ell,\theta}_{\text{node}};\ \underbrace{\underline{x}}_{\text{symbol in node}})$$

ℓth node group	θth node	$\underline{x}$th symbol
$\ell = 1, 2, \cdots, t$	$\theta \in \mathbb{F}_q$	$\underline{x} \in \mathbb{F}_q^t$

Parity Checks

Row-Sum Parity Checks:

$$\sum_{\ell=1}^{t} \sum_{\theta \in \mathbb{F}_q} C(\ell, \theta; \underline{z}) \quad = \quad 0$$

Jump (Zig-Zag) Parity Checks:

$$\sum_{\ell=1}^{t} \left(\sum_{\theta \neq z_\ell} C(\ell, \theta; \underline{z}) + C(\ell, z_\ell; \quad \underbrace{(\underline{z} - \Delta \underline{e}_\ell)}_{\text{jump in } \ell\text{th position}} \quad) \right) \quad = \quad 0$$

Illustrating Row-Sum Parity Checks ($z_1 = 0$ only)

$(x_1 x_2 x_3)$	$\ell = 1$		$\ell = 2$		$\ell = 3$	
	$\theta = 0$ Node 1	$\theta = 1$ Node 2	$\theta = 0$ Node 3	$\theta = 1$ Node 4	$\theta = 0$ Node 5	$\theta = 1$ Node 6
(000)	A	A	A	A	A	A
(001)	B	B	B	B	B	B
(010)	C	C	C	C	C	C
(011)	D	D	D	D	D	D
(100)						
(101)						
(110)						
(111)						

(A, B, C and D represent Row-Sum parity checks)

Illustrating Jump Parity Checks ($z_1 = 0$ only)

$(x_1x_2x_3)$	$\ell = 1$		$\ell = 2$		$\ell = 3$	
	$\theta = 0$ Node 1	$\theta = 1$ Node 2	$\theta = 0$ Node 3	$\theta = 1$ Node 4	$\theta = 0$ Node 5	$\theta = 1$ Node 6
(000)		P		P R		P Q
(001)		Q		Q S	P Q	
(010)		R	P R			R S
(011)		S	Q S		R S	
(100)	P					
(101)	Q					
(110)	R					
(111)	S					

- (P, Q, R and S represent Jump parity checks)
- From this it is clear how node 1 can be repaired by downloading 4 symbols from each of the other nodes

First refinement: Bringing in Coefficients

$$\sum_{\ell=1}^{t} \sum_{\theta \in \mathbb{F}_q} \underbrace{\lambda(\ell,\theta)}_{\text{coefficient}} \; C(\ell,\theta;\underline{z}) \quad = \quad 0$$

$$\sum_{\ell=1}^{t} \left(\sum_{\theta \neq z_\ell} \lambda(\ell,\theta) C(\ell,\theta;\underline{z}) + \lambda(\ell,z_\ell) C(\ell,z_\ell; \quad \underbrace{(\underline{z} - \Delta \underline{e}_\ell)}_{\text{jump in } \ell\text{th position}} \quad) \right) \quad = \quad 0$$

Second Refinement: Adding Extra Terms in the Parity Check Equations (for Data Collection)

$$\sum_{\ell=1}^{t} \sum_{\theta \in \mathbb{F}_q} \lambda(\ell,\theta) C(\ell,\theta;\underline{z}) \quad = \quad 0$$

$$\sum_{\ell=1}^{t} \left(\sum_{\theta \neq z_\ell} \lambda(\ell,\theta) C(\ell,\theta;\underline{z}) + \lambda(\ell,z_\ell) C(\ell,z_\ell; \underbrace{(\underline{z} - \Delta \underline{e}_\ell)}_{\text{jump in } \ell\text{th position}}) \right)$$

$$+ \quad \underbrace{\sum_{\ell=1}^{t} \sum_{\theta \in \mathbb{F}_q} \gamma(\ell,\theta) C(\ell,\theta;\underline{z})}_{\text{helps guarantee data-collection property}} \quad = \quad 0$$

Parity-Check Matrix (without extra terms)

Associated parity-check matrix H is of the form:

	$\ell = 1$				$\ell = 2$				$\ell = 3$			
	$\theta = 0$		$\theta = 1$		$\theta = 0$		$\theta = 1$		$\theta = 0$		$\theta = 1$	
	Node 1		Node 2		Node 3		Node 4		Node 5		Node 6	
$z_1 = 0$	I_4		I_4		I_4		I_4		I_4		I_4	
$z_1 = 1$		I_4		I_4		I_4		I_4		I_4		I_4
$z_1 = 0$		I_4	I_4		A_1		A_3		A_5		A_7	
$z_1 = 1$		I_4	I_4			A_2		A_4		A_6		A_8

- $\Delta = 0$ in the first two rows
- $\Delta = 1$ (indicating jump parity) in bottom two rows

Parity-Check Matrix (with extra terms in blue)

To ensure data recovery, replace H by the form:

$$H \quad = \quad H_0 + H_1$$

where H_0, H_1 are given respectively by:

	$\ell = 1$				$\ell = 2$				$\ell = 3$			
	$\theta = 0$		$\theta = 1$		$\theta = 0$		$\theta = 1$		$\theta = 0$		$\theta = 1$	
	Node 1		Node 2		Node 3		Node 4		Node 5		Node 6	
$z_1 = 0$	I_4		I_4		I_4		I_4		I_4		I_4	
$z_1 = 1$		I_4		I_4		I_4		I_4		I_4		I_4
$z_1 = 0$	I_4		I_4		I_4		I_4		I_4		I_4	
$z_1 = 1$		I_4		I_4		I_4		I_4		I_4		I_4

	$\ell = 1$				$\ell = 2$				$\ell = 3$			
	$\theta = 0$		$\theta = 1$		$\theta = 0$		$\theta = 1$		$\theta = 0$		$\theta = 1$	
	Node 1		Node 2		Node 3		Node 4		Node 5		Node 6	
$z_1 = 0$												
$z_1 = 1$												
$z_1 = 0$		I_4	I_4		A_1		A_3		A_5		A_7	
$z_1 = 1$		I_4	I_4			A_2		A_4		A_6		A_8

(this ensures the data collection property; Polynomial root counting)

Codes with Locality

Some References

1. Gopalan, Huang, Yekhanin, Simitci, T-IT, Nov. 2012, winner of joint COMSOC-IT Best Paper Award.
2. P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the Locality of Codeword Symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.
3. M. Forbes and S. Yekhanin, "On the locality of codeword symbols in non-linear codes", arXiv:1303:3921, 2013.
4. C. Huang, M. Chen, J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," *Sixth IEEE International Symposium on Network Computing and Applications,* 2007.
5. J. Han and L. A. Lastras-Montano, "Reliable memories with subline accesses," *Proc. IEEE Internat. Sympos. Inform. Theory*, 2007, pp. 2531-2535.
6. D. S. Papailiopoulos, A. G. Dimakis, "Locally repairable codes," *ISIT,* 2012.
7. F. Oggier, A. Datta, "Self-repairing homomorphic codes for distributed storage systems," *IEEE INFOCOM,* 2011.
8. D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang, and J. Li, "Simple regenerating codes: Network coding for cloud storage, " *Proc. IEEE INFOCOM*, 2012, pp. 2801-2805.

United by an Acronym

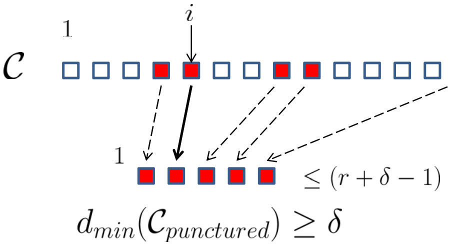
$$
\begin{aligned}
\text{Codes with Locality} &\equiv \text{locally repairable codes} \\
&\equiv \text{locally recoverable codes} \\
&\equiv \text{locally reconstructible codes} \\
&\equiv \text{local reconstruction codes} \\
&\equiv \text{LRC} \quad !
\end{aligned}
$$

Codes with Locality

Setting: $\mathcal{C}$ is an $[n, \kappa, d_{\min}]$ linear code. $\{c_i\}_{i=1}^n$ are code symbols.

- Code symbol c_j has *locality* (r, δ) if there exists a subset of code symbols $\{c_1, \cdots, c_n\}$ that includes c_j and forms a "local" code with parameters:

$$[\text{length} \leq r + \delta - 1, \quad \text{dimension} \leq r, \quad d_{\min} \geq \delta]$$



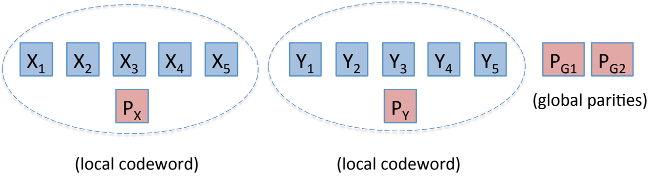
All-Symbol and Information-Symbol Locality (r, δ)

$$\text{Codewords in } \mathcal{C} : \quad \left(\underbrace{c_1, c_2, \ldots c_k}_{\text{information set}}, c_{k+1}, c_{k+2}, \ldots, c_n \right),$$

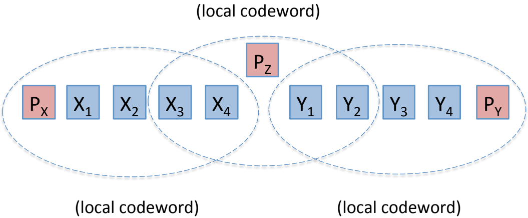
- $\{c_j\}_{j=1}^k$ is an information set if message symbols can be uniquely decoded from $\{c_j\}_{j=1}^k$, but not from any subset of $\{c_j\}_{j=1}^k$
- $\mathcal{C}$ is said to have information symbol locality (r, δ), if all k code symbols comprising an information set $\{c_j\}_{j=1}^k$ have locality (r, δ)

- Code $\mathcal{C}$ is said to have all-symbol locality (r, δ), if all n code symbols have (r, δ) locality

Illustrating Information and All-Symbol Locality



(information-symbol locality)



(all-symbol locality)

Bound on Global Minimum Distance

Theorem

If an $[n, \kappa, d_{\min}]$ code $\mathcal{C}$ has information symbol locality (r, δ), then

$$d_{\min} \leq \underbrace{(n - \kappa + 1)}_{\textit{Singleton bound}} - \underbrace{\left(\left\lceil \frac{\kappa}{r} \right\rceil - 1 \right) (\delta - 1)}_{\textit{loss due to locality}} .$$

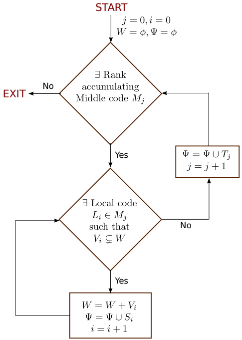
- Bound established by P. Gopalan et al. for the case when the local codes are parity check codes ($\delta = 2$)
- Our extension to the general case is straightforward, but useful

---

- Gopalan, Huang, Yekhanin, Simitci, T-IT, Nov. 2012.
- Prakash, Kamath, Lalitha, and PVK, (ISIT 2012), Jul. 2012.

Derivation of the Bound on Minimum Distance

- Based on a recursive algorithm that searches for a large $(k \times \ell)$ sub-matrix of the generator matrix whose rank is $\leq (k-1)$.
- keep adding columns of G while slowing rank increase in matrix



$$G = \left[\begin{array}{|cccc|ccc} g_{11} & g_{12} & g_{13} & g_{1\ell} & g_{15} & g_{16} & g_{17} \\ g_{21} & g_{22} & g_{23} & g_{2\ell} & g_{25} & g_{26} & g_{27} \\ g_{31} & g_{32} & g_{33} & g_{3\ell} & g_{35} & g_{36} & g_{37} \end{array}\right]$$

($\ell = 4$ here) Then we have:

$$d_{min} \leq (n - \ell).$$

- P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the Locality of Codeword Symbols," *IEEE Trans. Inf. Theory*, Nov. 2012.

Pyramid Codes: Codes with Optimal Information-Symbol Locality

- Given generator matrix G of a systematic $[7, 4, 4]$ MDS code:

$$G = \begin{bmatrix} 1 & & & & g_{11} & g_{12} & g_{13} \\ & 1 & & & g_{21} & g_{22} & g_{23} \\ & & 1 & & g_{31} & g_{32} & g_{33} \\ & & & 1 & g_{41} & g_{42} & g_{43} \end{bmatrix}$$

- Split first two "parity" columns, and then rearrange columns:

$$\left[\begin{array}{cccc|cccc|c} 1 & & g_{11} & g_{12} & & & & & g_{13} \\ & 1 & g_{21} & g_{22} & & & & & g_{23} \\ \hline & & & & 1 & & g_{31} & g_{32} & g_{33} \\ & & & & & 1 & g_{41} & g_{42} & g_{43} \end{array}\right]$$

The new $[9, 4, 4]$ code has two $[4, 2, 3]$ local codes and is optimal.

C. Huang, M. Chen, and J. Li " Pyramid Codes: Flexible Schemes to Trade Space for Access Efficiency in Reliable Data Storage Systems," *NCA 2007*.

Some Optimal Constructions of Codes with Locality

Explicit Constructions

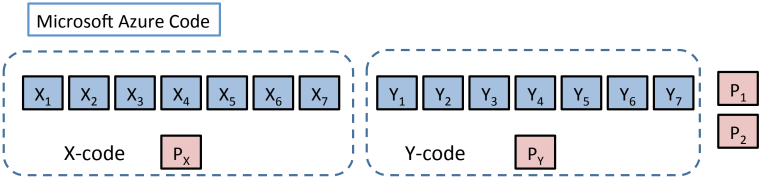
1. Pyramid Code construction for information locality.
2. Parity splitting construction for all symbol locality: $n = \left\lceil \frac{k}{r} \right\rceil (r + \delta - 1)$.
3. Rank-Distance based code with all-symbol locality : $\delta = 2$.
4. Tamo-Barg construction

Non-Explicit Construction All symbol locality codes can be constructed whenever

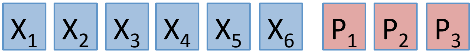
$$(r + \delta - 1) | n, \text{ provided } q > \binom{n-1}{k-1}$$

- C. Huang, M. Chen, and J. Li "Pyramid Codes: Flexible Schemes to Trade Space for Access Efficiency in Reliable Data Storage Systems," *NCA 2007*.
- J. Han, L. A. Lastras-Montano; , "Reliable Memories with Subline Accesses," *ISIT*- 2007.
- P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the Locality of Codeword Symbols," IT-Trans, Nov. 2012.
- N. Prakash, G. M. Kamath, V. Lalitha, and PVK, "Optimal linear codes with a local-error-correction property," *ISIT*-2012.
- N. Silberstein, A. S. Rawat and S. Vishwanath, "Error Resilience in Distributed Storage via Rank-Metric Codes", *Allerton*, 2012.
- Itzhak Tamo and Alexander Barg, "A Family of Optimal Locally Recoverable Codes," T-IT, Aug 2014. (IT-Trans. best paper award).

Windows Azure Storage Coding Solution



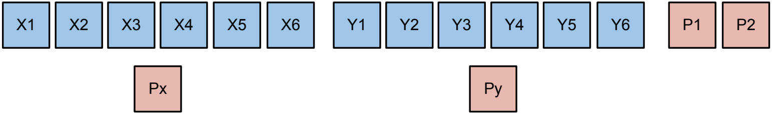
Comparison: In terms of reliability and number of helper nodes contacted for node repair, the two codes are comparable. The overheads however are quite different, 1.29 for the Azure code versus 1.5 for the RS code. This difference has reportedly saved Microsoft millions of dollars.



Huang, Simitci, Xu, Ogus, Calder, Gopalan, Li, Yekhanin, "Erasure Coding in Windows Azure Storage," USENIX, Boston, MA, 2012.

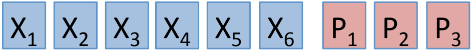
Windows Azure Storage Coding Solution (continued)

Windows Azure Code:

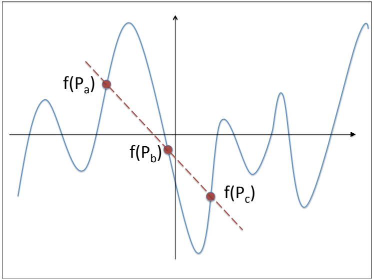


Comparison: In terms of reliability and number of helper nodes contacted for node repair, the two codes are comparable. The overheads however are quite different, 1.33 for the Azure code versus 1.5 for the RS code. This difference has reportedly saved Microsoft millions of dollars.

Reed-Solomon Code



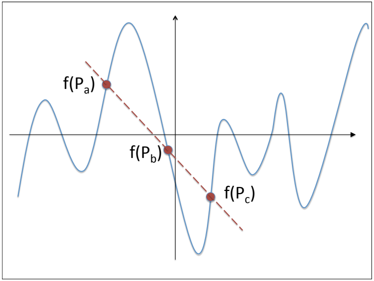
The Tamo-Barg Construction (all-symbol locality)



- subset of RS codewords: $(f(P_1), f(P_2), \cdots, f(P_n))$, with $\deg(f) \leq (k-1)$
- subset ensures that given point P_a there exist other points fitted by a lower degree polynomial which can be used for correction
- for example, to a line when evaluated at 3 points; this provides locality
- provides low-field-size constructions for many parameter sets

Itzhak Tamo and Alexander Barg, "A Family of Optimal Locally Recoverable Codes," T-IT, Aug. 2014, . (IT-Trans. best paper award).

The Tamo-Barg Construction

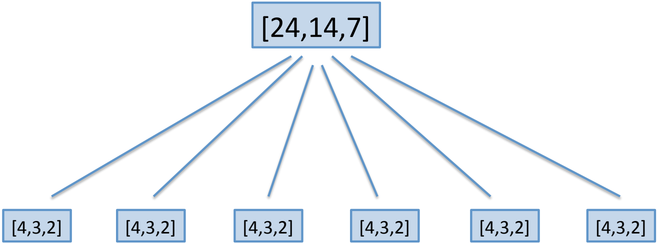


- subset of RS codewords: $(f(P_1), f(P_2), \cdots, f(P_n))$, with $\deg(f) \leq (k-1)$
- subset ensures that given point P_a there exist other points fitted by a lower degree polynomial which can be used for correction
- for example, to a line when evaluated at 3 points; this provides locality
- provides low-field-size constructions for many parameter sets
- There is also a Chinese Remainder Theorem interpretation

Codes with Hierarchical Locality

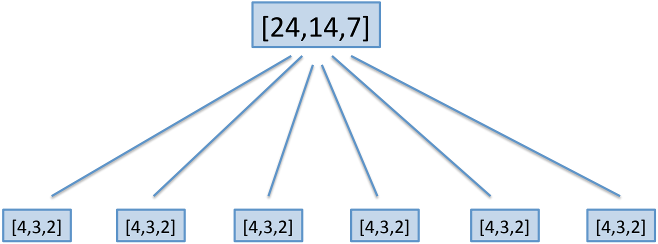
Birenjith Sasidharan, Gaurav Kumar Agarwal, PVK, “Codes With Hierarchical Locality,” ISIT 2015.

Codes with Locality



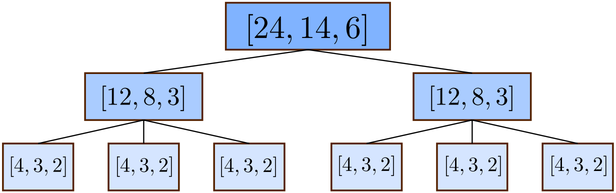
$$
\begin{aligned}
d &\leq \underbrace{(n-k+1)}_{\text{Singleton bound}} - \underbrace{\left(\lceil \frac{k}{r} \rceil - 1\right)(\delta - 1)}_{\text{loss due to locality}} \\
r &= \text{locality} \\
\delta &= \text{minimum distance of the local code}
\end{aligned}
$$

Codes with Locality do not Scale



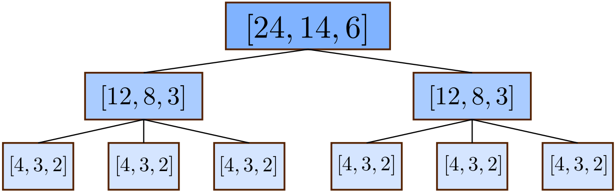
- If the local code is overwhelmed, then one has to appeal to the overall code which means contacting all 14 nodes for node repair.
- Is it possible to build a code where the repair degree increases gradually as opposed to in a single jump ?

Codes with Hierarchical Locality



- Codes with hierarchical locality do exactly that by calling for help from an intermediate layer of codes when the local code fails.
- These codes may be regarded as the “middle codes”.

Codes with Hierarchical Locality - Parameters

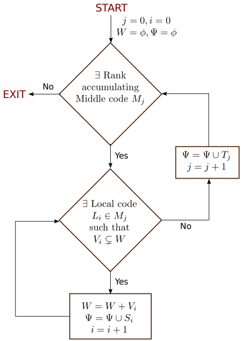


$$d \leq \underbrace{n - k + 1 - \left(\left\lceil \frac{k}{r_2} \right\rceil - 1\right)(\delta_2 - 1)}_{\text{bound for codes with locality}} - \underbrace{\left(\left\lceil \frac{k}{r_1} \right\rceil - 1\right)(\delta_1 - \delta_2)}_{\text{additional loss for 2nd locality layer}}$$

Derivation of the Bound on Minimum Distance

- Proceeds along the lines of the original paper on codes with locality
- Based on a recursive algorithm that searches for a large $(k \times \ell)$ sub-matrix of the generator matrix whose rank is $\leq (k-1)$.

$$G = \left[\left\| \begin{array}{cccc} g_{11} & g_{12} & g_{13} & g_{1\ell} \\ g_{21} & g_{22} & g_{23} & g_{2\ell} \\ g_{31} & g_{32} & g_{33} & g_{3\ell} \end{array} \right\| \begin{array}{ccc} g_{15} & g_{16} & g_{17} \\ g_{25} & g_{26} & g_{27} \\ g_{35} & g_{36} & g_{37} \end{array} \right]$$

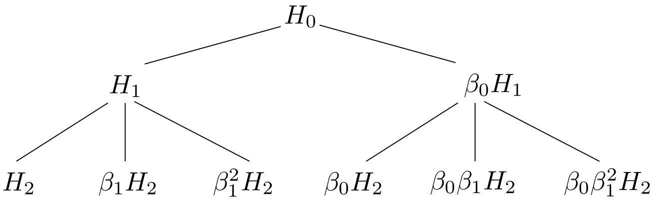


$(\ell = 4$ here) Then we have:

$$d_{min} \leq (n - \ell).$$

All-symbol Local Optimal Construction: An Example

- Need to satisfy a divisibility condition $n_2 \mid n_1 \mid n$
- Example: $[24, 14]$, $[12, 8]$, $[4, 3]$. Here: $(n_2 = 4 \mid n_1 = 12 \mid n = 24)$.



1. Choose $\mathbb{F}_{25}$.
2. Identify subgroup chain $H_2 \subseteq H_1 \subseteq H = \mathbb{F}_{25}^*$
3. Coset decomposition - supports of local codes

A ChineseRemainder-Theorem-Based All-symbol Local Optimal Construction

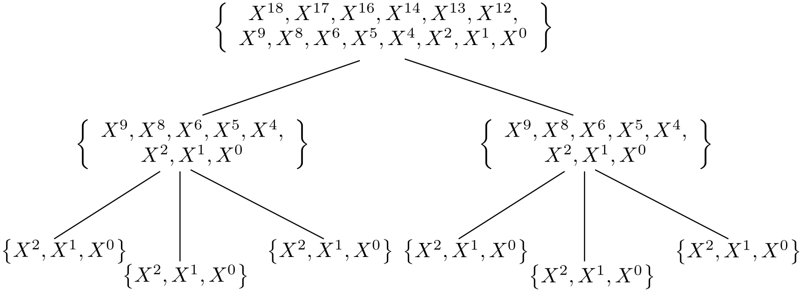
$$\left\{ \begin{array}{c} X^{22}, X^{21}, X^{20}, X^{18}, X^{17}, X^{16}, X^{14}, X^{13}, X^{12}, \\ X^{10}, X^{9}, X^{8}, X^{6}, X^{5}, X^{4}, X^{2}, X^{1}, X^{0} \end{array} \right\}$$

$$\left\{ \begin{array}{c} X^{10}, X^{9}, X^{8}, \\ X^{6}, X^{5}, X^{4}, \\ X^{2}, X^{1}, X^{0} \end{array} \right\} \qquad \left\{ \begin{array}{c} X^{10}, X^{9}, X^{8}, \\ X^{6}, X^{5}, X^{4}, \\ X^{2}, X^{1}, X^{0} \end{array} \right\}$$

$$\{X^2, X^1, X^0\} \quad \{X^2, X^1, X^0\} \quad \{X^2, X^1, X^0\} \quad \{X^2, X^1, X^0\} \quad \{X^2, X^1, X^0\} \quad \{X^2, X^1, X^0\}$$

- The tree above shows the monomials appearing in the restriction of the code polynomial (its monomials appear on top) to each local code.

All-symbol Local Optimal Construction: An Example (continued)



- The local codes can be tied together using an overall global code by simply restricting the set of code polynomials at the top. Here we do not allow the maximum degree to exceed 18. (The maximum was previously 22).

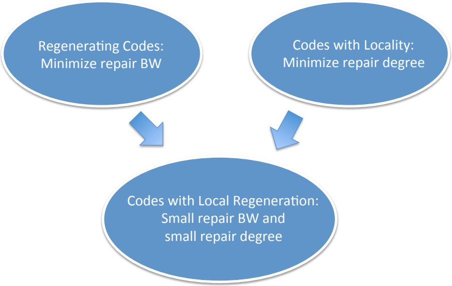
Codes with Local Regeneration

Some References

1. A. S. Rawat, O. O. Koyluoglu, N. Silberstein, S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," T-IT, Jan 2014.
2. G. M. Kamath, N. Prakash, V. Lalitha, PVK, 'Codes With Local Regeneration and Erasure Correction," T-IT, Aug. 2014 .
3. N. Prakash, G.M. Kamath, V. Lalitha, PVK, A.S. Rawat, O.O. Koyluoglu, N. Silberstein, S. Vishwanath, "Explicit MBR All-Symbol Locality Codes,"ISIT 2013.
4. M. N. Krishnan, N. Prakash, V. Lalitha, B. Sasidharan, PVK, S. Narayanamurthy, R. Kumar and S. Nandi, "Evaluation of codes with inherent double replication for Hadoop", in *Proc. USENIX HotStorage,* 2014

(first two references represent independent work carried out in parallel
the last reference is to an evaluation through hardware emulation in collaboration with NetApp)

Codes with Local Regeneration

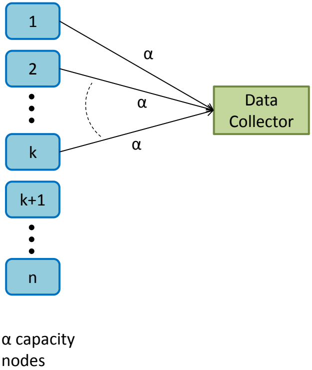


- Combine notions of locality and low-bandwidth regeneration
- New upper bounds on minimum distance
- optimal code constructions

---

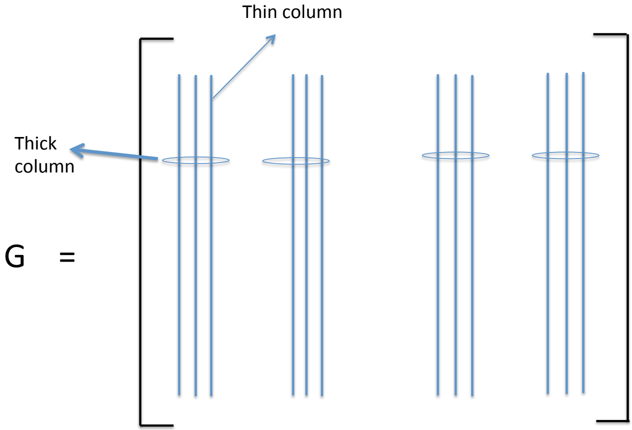
1. G. M. Kamath, N. Prakash, V. Lalitha, PVK, 'Codes With Local Regeneration and Erasure Correction," T-IT, Aug. 2014 .

Vector Code Viewpoint



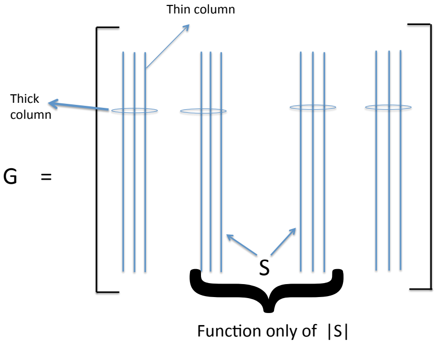
Regenerating codes can be viewed as codes over the vector alphabet $\mathbb{F}_q^\alpha$ since each node stores α symbols.

Generator Matrix of a Vector Code - Thin and Thick Columns



Here $\alpha = 3$, so there are 3 thin columns per thick column

Codes with Uniform Rank Accumulation



- If $\mathcal{C}$ has length n, then G will have n thick columns.
- Let S be any subset consisting of $|S|$ thick columns.
- Then $\mathcal{C}$ has the uniform rank accumulation (URA) property if

$$\mathrm{Rank}(G|_S)$$

is a function of $|S|$ alone.

Examples of Codes with Uniform Rank Accumulation

Set

$$\begin{aligned} b_i &= \text{Rank}(G|_S), \quad |S| = i \\ a_i &= b_i - b_{i-1}, \; i \geq 1 \qquad \text{(incremental rank)} \end{aligned}$$

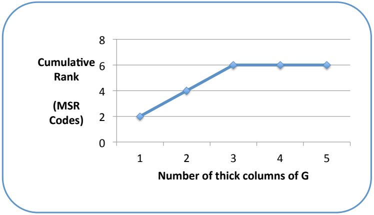
$$b_j = \sum_{i=1}^{j} a_i \qquad \text{(cumulative rank).}$$

Then $a_1 \geq a_2 \geq \cdots \geq a_n$

- A scalar code has the URA property iff it is an MDS code
- Both MSR and MBR codes have the URA property
- there are other examples as well...

Uniform Rank Accumulation – MSR Code

URA profile of an $(n = 5, k = 3, d = 4)$, $(\alpha = 2, \beta = 1)$ MSR Code



$$(a_1, a_2, a_3, a_4, a_5) \quad = \quad (2, 2, 2, 0, 0)$$

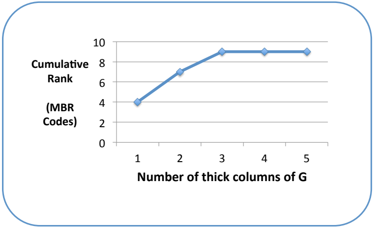
Cumulative rank:

$$b_j \quad = \quad \sum_{i=1}^{j} a_i.$$

$$(b_1, b_2, b_3, b_4, b_5) \quad = \quad (2, 4, 6, 6, 6)$$

Uniform Rank Accumulation – MBR Code

URA profile of an $(n = 5, k = 3, d = 4)$, $(\alpha = 4, \beta = 1)$ MBR Code



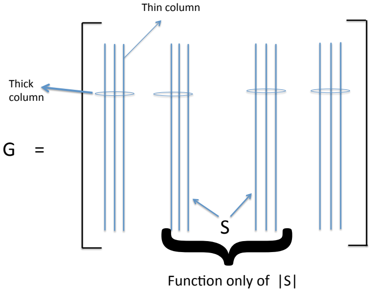
$$(a_1, a_2, a_3, a_4, a_5) = (4, 3, 2, 0, 0)$$

Cumulative rank:

$$b_j = \sum_{i=1}^{j} a_i.$$

$$(b_1, b_2, b_3, b_4, b_5) = (4, 7, 9, 9, 9)$$

Back to Vector Codes with Uniform Rank Accumulation



Recall that a vector code has the URA has the uniform rank accumulation (URA) property if

$$\mathrm{Rank}(G|_S) \quad = \quad b_{|S|}.$$

The Cumulative Function P

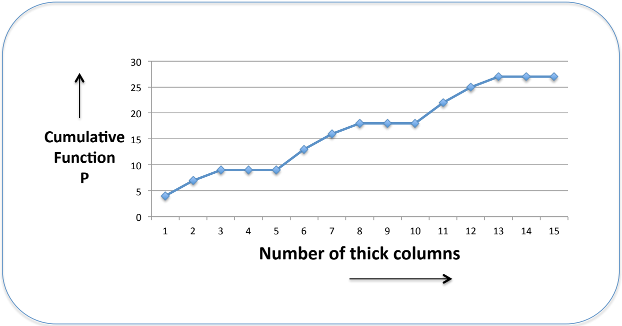
Let the sequence $\{a_j\}$ be repeated periodically:

$$(\; a_1, a_2, \cdots a_n, \quad a_1, a_2, \cdots a_n, \quad a_1, a_2, \cdots, a_n \quad \cdots \;)$$

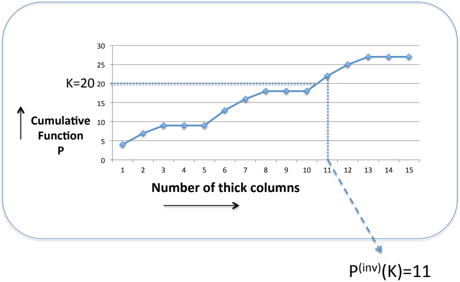
$$P(j) \;=\; \text{sum of first } j \text{ terms of this periodic sequence..}$$

Can be verified that $P(\cdot)$ is sub-additive:

$$P(x+y) \;\leq\; P(x) + P(y).$$



The $d_{\min}$ Bound When the Local Codes have URA



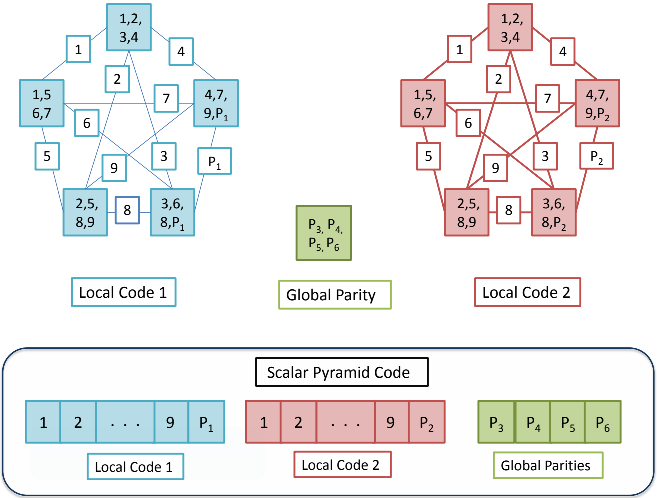
Let S be maximal w.r.t. $\text{Rank}(G|_S) < K$ where $K = \text{Rank}(G)$. Then

$$\begin{aligned} d_{\min} &= n - |S| = n - (P^{(\text{inv})}(K) - 1) \quad \text{where} \\ P^{(\text{inv})}(K) &= j \text{ if } P(j-1) < K \leq P(j). \end{aligned}$$

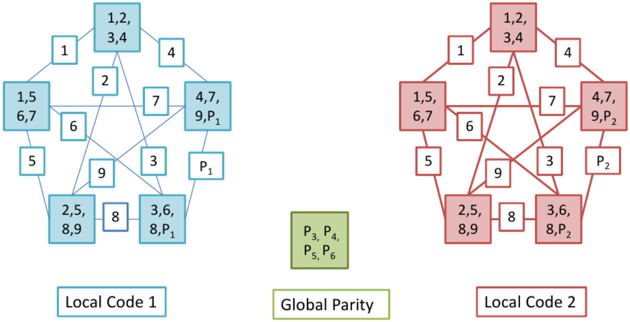
In the example, $d_{\min} \leq 15 - (11 - 1) = 5$.

(Global) Code with MBR Code Locality

The construction makes use of the scalar pyramid code and is optimal:



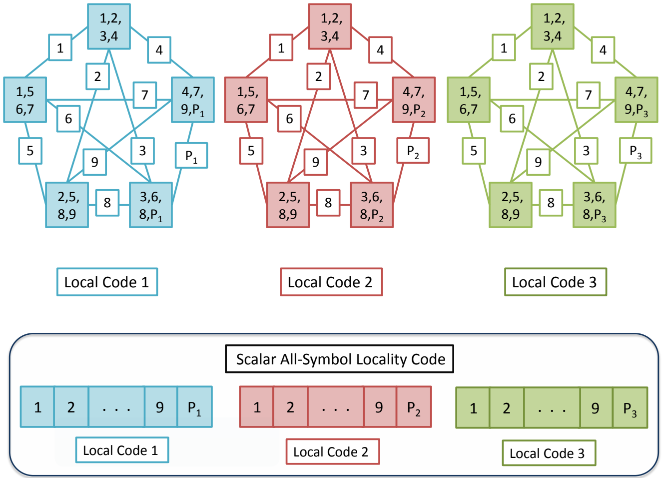
Performance in Terms of Repair BW and Repair "Degree"



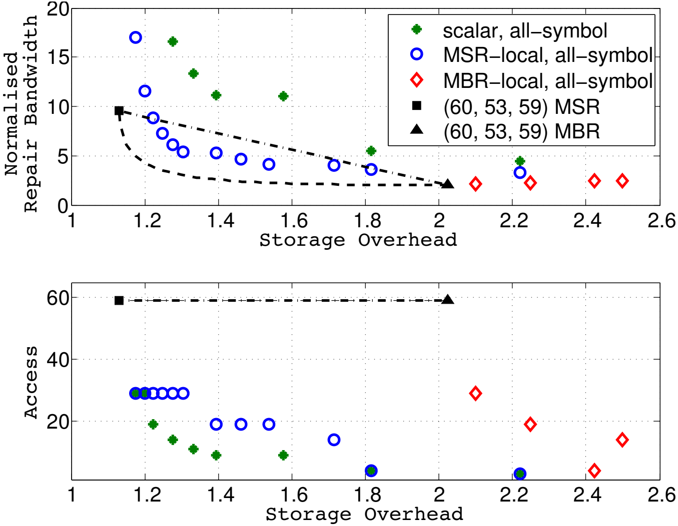
- Global Code: length $= 11$, $d_{\text{min}} = 4$,
- Local MBR Codes: length $(r + \delta - 1) = 5$, minimum distance $\delta = 3$,
- Local MBR codes are optimum in terms of repair
- Repair degree $= 4$ through locality.

(Global) Code with AS-MBR Code Locality

The construction makes can make use of an all-symbol local scalar code and is also optimal:



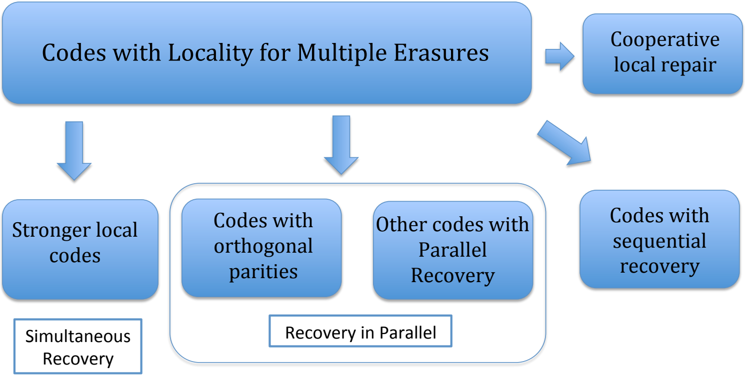
Code Comparison Based on Repair BW, Repair Degree for Given Storage Overhead



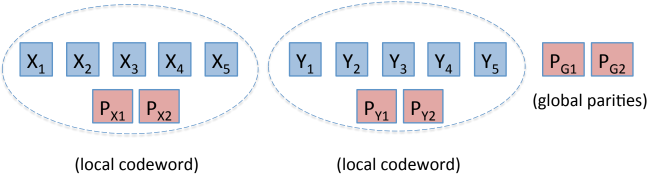
Codes with Locality for Multiple Erasures

Different Approaches: Codes with Locality for Multiple Erasures

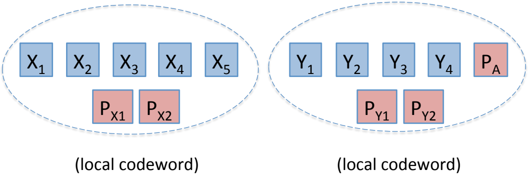
- Increasing trend towards low-cost commodity servers with higher failure rates
- Presence of "hot" nodes which are inaccessible during repair



Handling Multiple Erasures: Stronger Local Codes Approach



(Information-Symbol locality)



More on 'Stronger Local Codes Approach'

If an $[n, \kappa, d_{\min}]$ code $\mathcal{C}$ has information symbol locality r, then

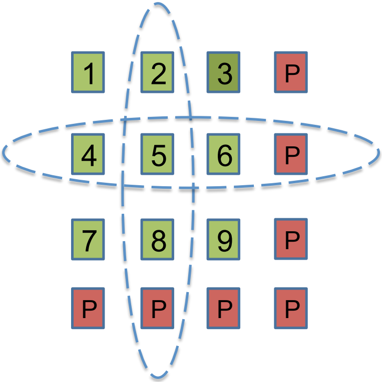
$$\boxed{d_{\min} \leq \underbrace{(n - \kappa + 1)}_{\text{Singleton bound}} - \underbrace{\left(\left\lceil \frac{\kappa}{r} \right\rceil - 1\right)(\delta - 1)}_{\text{price for locality requirement}} .}$$

- Generalization of the Gopalan et al bound
- Pyramid code construction can be extended to this case as can the construction by Tamo and Barg
- More recent results by Wentu Song, Son Hoang Dau, Chau Yuen, and Tiffany Jing Li

---

- N. Prakash, G. Kamath, V. Lalitha, and PVK, "Optimal linear codes with a local-error-correction property," in ISIT 2012.
- Optimal Locally Repairable Linear Codes, by Wentu Song, Son Hoang Dau, Chau Yuen, and Tiffany Jing Li.

Example of the Orthogonal Parity-Check Approach



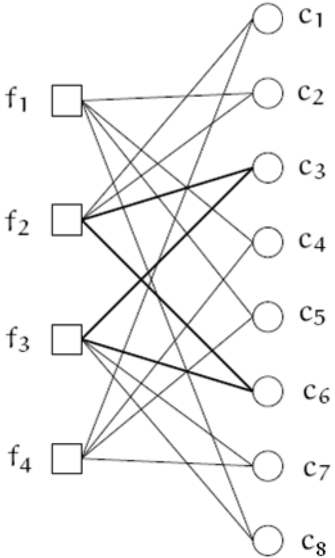
- Each data symbol is protected by two local codes with disjoint support
- All local codes are single-parity-check codes

LDPC Code Connection

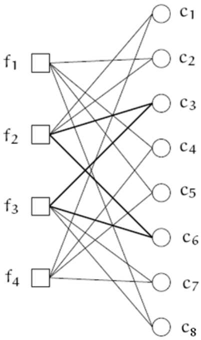
Codes with orthogonal parity-checks can also be obtained from (d_v, d_c)-regular LDPC Codes, assuming the absence of cycles of length ≤ 4.

(this is well known)

An Example (d_v, d_c)-Regular LDPC Code



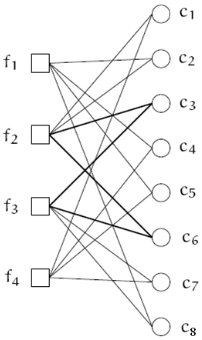
An Example (d_v, d_c)-Regular LDPC Code



Our interest is in those codes where

- each variable node has degree t
- each check node has degree $(r + 1)$
- there are no cycles of length 4

An Example (d_v, d_c)-Regular LDPC Code

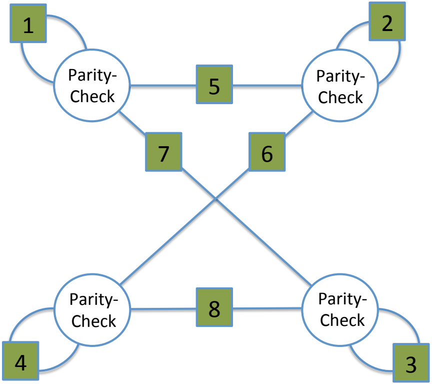


This ensures that:

- each code symbol has locally r
- Each code symbol is protected by t orthogonal parity checks

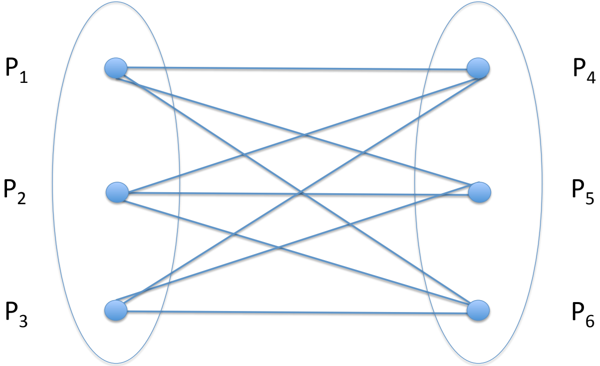
Codes for Two-Erasure Correction

The Sequential-Recovery Approach - An Example



The Sequential-Recovery Approach - A More General Turan-Graph Framework

Turan Graph



(9 edges form remaining code symbols)
(n=15)

The Sequential-Recovery Approach - A More General Turan-Graph Framework

- The Turan graph construction has an additional feature that it leads to optimal solutions for smaller rates than the rate that arises from the constraints
- This can be explained using the theory of Generalized Hamming Weights of a block code

---

- V.K. Wei, "Generalized Hamming Weights for Linear Codes," IEEE Trans. Inform. Th, 1991.

Thanks!

Additional References

Note: This list adds to the papers referenced in the slides. Coding for distributed storage is a rapidly growing field of research activity and there are a large and ever-growing number of publications in this area. The listing below does not claim in any way to be comprehensive, and apologies are offered in advance for any missing references.

Regenerating Codes

1. A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory,* vol. 56 no. 9, pp. 4539–4551, 2010.
2. Y. Wu, A. G. Dimakis, K. Ramchandran, "Deterministic Regenerating Codes for Distributed Storage," *45th Annual Allerton Conference on Communication, Control, and Computing,* Allerton, 2007.

Regenerating Codes - MSR and MBR Constructions

1. V.R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of MDS codes in distributed storage," *IEEE Transactions on Information Theory*, vol. 59 no. 5 pp. 2974–2987, 2013.
2. K. V. Rashmi, N. B. Shah, P.V. Kumar, "Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction," *IEEE Transactions on Information Theory*, vol. 57, no.8, pp. 5227–5239, Aug. 2011. 8
3. N. B. Shah, K. V. Rashmi, P. V. Kumar, K. Ramchandran, "Distributed Storage Codes With Repair-by-Transfer and Nonachievability of Interior Points on the Storage-Bandwidth Tradeoff," *IEEE Transactions on Information Theory*, vol. 58 no. 3, pp. 1837–1852, March 2012.
4. N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference Alignment in Regenerating Codes for Distributed Storage: Necessity and Code Constructions," *IEEE Transactions on Information Theory,* vol. 58, no. 4, pp. 2134–2158, April 2012.
5. C. Suh and K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *IEEE Transactions on Information Theory,* vol. 57, no. 3, pp. 1425–1442, 2011.
6. I. Tamo, Z. Wang, J. Bruck, "Zigzag Codes: MDS Array Codes with Optimal Rebuilding, *CoRR,* abs/1112.0371, 2011.
7. D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair Optimal Erasure Codes through Hadamard Designs," *IEEE Transactions on Information Theory,* vol. 59, no.5, pp. 3021–3037, 2013.
8. Z. Wang, I. Tamo, J. Bruck, "On Codes for Optimal Rebuilding Access," *Allerton*, 2011.

Regenerating Codes - MSR and MBR Constructions (contd.)

1. K. Rashmi, N. Shah, P. Kumar and K. Ramchandran, “Explicit construction of optimal exact regenerating codes for distributed storage,” in *Proc. 47th Annu. Allerton Conf. Communication, Control, and Computing*, Urbana-Champaign, IL, Sep. 2009, pp. 1243–1249
2. S. Lin and W. Chung, “Novel Repair-by-Transfer Codes and Systematic Exact-MBR Codes with Lower Complexities and Smaller Field Sizes,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3232–3241, 2014
3. B. Sasidharan, G. K. Agarwal and P. V. Kumar, ”A high-rate MSR code with polynomial sub-packetization level,” *Information Theory (ISIT), IEEE International Symposium on,* 2015, pp. 2051-2055
4. Y. S. Han, H. Pai, R. Zheng, and P. K. Varshney, “Update-Efficient Error- Correcting Product-Matrix Codes,” *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 19251938, Jun. 2015
5. G. K. Agarwal, B. Sasidharan and P. V. Kumar, “An alternate construction of an access-optimal regenerating code with optimal sub-packetization level”, in *Proc. Communications (NCC), National Conference on,* 2015
6. A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, “Progress on High-rate MSR Codes: Enabling Arbitrary Number of Helper Nodes,” *CoRR,* abs/1601.06362, 2016
7. M. N. Krishnan and P. V. Kumar, “On MBR codes with replication,” *CoRR,* abs/1601.08190, 2016
8. S.Goparaju, A. Fazeli and A. Vardy, “Minimum Storage Regenerating Codes For All Parameters,” *CoRR,* abs/1602.04496, 2016
9. N. Raviv, N. Silberstein and T. Etzion, “Constructions of High-Rate Minimum Storage Regenerating Codes over Small Fields,” *CoRR,* abs/1505.00919

Interior-Point Constructions (Between MSR and MBR Points)

1. S. Goparaju, S. El Rouayheb, A. R. Calderbank, "New Codes and Inner Bounds for Exact Repair in Distributed Storage Systems," ISIT 2014
2. T. Ernvall, "Exact-Regenerating Codes between MBR and MSR Points," CoRR abs/1304.5357
3. B. Sasidharan, P. V. Kumar, "High-rate regenerating codes through layering," ISIT 2013.
4. C. Tian, B. Sasidharan, V. Aggarwal, V. A. Vaishampayan, and P. V. Kumar, "Layered exact-repair regenerating codes via embedded error correction and block designs," *IEEE Trans. Inf. Theory,* vol. 61, no. 4, pp. 1933–1947, 2015.
5. K. Senthoor, B. Sasidharan and P. V. Kumar, "Improved layered regenerating codes characterizing the exact-repair storage-repair bandwidth tradeoff for certain parameter sets," in *Proc. IEEE Information Theory Workshop (ITW),* 2015.

Exact-Repair Tradeoff

1. N. B. Shah, K. V. Rashmi, P. V. Kumar, K. Ramchandran, "Distributed Storage Codes With Repair-by-Transfer and Nonachievability of Interior Points on the Storage-Bandwidth Tradeoff," *IEEE Transactions on Information Theory*, vol. 58 no. 3, pp. 1837–1852, March 2012.
2. C. Tian, Characterizing the rate region of the (4, 3, 3) exact-repair regenerating codes, *IEEE Journal on Selected Areas in Communications,* vol. 32, no. 5, pp. 967975, May 2014.
3. B. Sasidharan, K. Senthoor, P. V. Kumar, "An Improved Outer Bound on the Storage-Repair-Bandwidth Tradeoff of Exact-Repair Regenerating Codes." ISIT 2014
4. N. Prakash and M. N. Krishnan, "The storage-repair-bandwidth trade-off of exact repair linear regenerating codes for the case d = k = n - 1," in *Proc. IEEE International Symposium on Information Theory, ISIT,* 2015.
5. S. Mohajer and R. Tandon, "New bounds on the (n, k, d) storage systems with exact repair," in *Proc. IEEE International Symposium on Information Theory, ISIT*, 2015
6. M. Elyasi, S. Mohajer and R. Tandon, "Linear exact repair rate region of (k + 1, k, k) distributed storage systems: A new approach," in *Proc. IEEE International Symposium on Information Theory, ISIT*, 2015
7. I. M. Duursma, " Outer bounds for exact repair codes," CoRR abs/1406.4852
8. I. M. Duursma, "Shortened regenerating codes," CoRR abs/1505.00178

Sub-Packetization Bounds

1. S. Goparaju, I. Tamo, and R. Calderbank, "An Improved SubPacketization Bound for Minimum Storage Regenerating Codes," in *IEEE Transactions on Information Theory*, vol. 60, no. 5, 2014, pp. 2770–2779.
2. I. Tamo, Z. Wang, J. Bruck, "Access vs. bandwidth in codes for storage," *ISIT*, 2012.

Cooperative and Adaptive Repair

1. K. W. Shum, Y. Hu, "Cooperative Regenerating Codes," *CoRR* abs/1207.6762, 2012.
2. A. M. Kermarrec, N. Le Scouarnec, G. Straub, "Repairing Multiple Failures with Coordinated and Adaptive Regenerating Codes," *NetCod*, 2011.
3. A. Wang, Z. Zhang, "Exact Cooperative Regenerating Codes with Minimum-Repair-Bandwidth for Distributed Storage," *INFOCOM*, 2013.

Fractional-Repetition Codes

1. N. Silberstein and T. Etzion, "Optimal fractional repetition codes based on graphs and designs," *IEEE Trans. Inf. Theory,* vol. 61, no. 8, pp. 41644180, 2015.
2. J. C. Koo and J. T. Gill. III, "Scalable constructions of fractional repetition codes in distributed storage systems," in *Proc. 49th Annual Allerton Conf. on Comm., Control, and Computing,* pp. 1366–1373, 2011.
3. O. Olmez and A. Ramamoorthy, "Repairable replication-based storage systems using resolvable designs," in *Proc. 50th Annual Allerton Conf. on Comm., Control, and Computing,* pp. 1174–1181, 2012.
4. S. El Rouayheb, K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," *Allerton*, 2010.

Codes with Locality

1. P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the Locality of Codeword Symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.
2. M. Forbes and S. Yekhanin, "On the locality of codeword symbols in non-linear codes", arXiv:1303:3921, 2013.
3. C. Huang, M. Chen, J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," *Sixth IEEE International Symposium on Network Computing and Applications*, 2007.
4. J. Han and L. A. Lastras-Montano, "Reliable memories with subline accesses," *Proc. IEEE Internat. Sympos. Inform. Theory*, 2007, pp. 2531-2535.
5. D. S. Papailiopoulos, A. G. Dimakis, "Locally repairable codes," *ISIT*, 2012.
6. F. Oggier, A. Datta, "Self-repairing homomorphic codes for distributed storage systems," *IEEE INFOCOM*, 2011.
7. D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang, and J. Li, "Simple regenerating codes: Network coding for cloud storage, " *Proc. IEEE INFOCOM*, 2012, pp. 2801-2805.
8. N. Prakash, G.M. Kamath, V. Lalitha, P. V. Kumar, "Optimal linear codes with a local-error-correction property," *ISIT*, 2012.
9. I. Tamo, A. Barg, "A Family of Locally Recoverable Codes," ISIT 2014.

Codes Combining Regeneration with Locality

1. A. S. Rawat, O. O. Koyluoglu, N. Silberstein, S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Transactions on Information Theory*, Jan. 2014.
2. G. M. Kamath, N. Prakash, V. Lalitha, P.V. Kumar, "Codes with local regeneration," ISIT 2013, also *IEEE Transactions on Information Theory*, Aug. 2014.
3. N. Prakash, G.M. Kamath, V. Lalitha, P.V. Kumar, A.S. Rawat, O.O. Koyluoglu, N. Silberstein, S. Vishwanath, "Explicit MBR All-Symbol Locality Codes,"ISIT 2013.

Maximal Recoverable and Partial-MDS Codes

1. M. Chen, C. Huang, and J. Li, "On the Maximally Recoverable Property for Multi-Protection Group Codes", ISIT 2007.
2. P. Gopalan, C. Huang, B. Jenkins, S. Yekhanin, "Explicit maximally recoverable codes with locality," *CoRR*, abs/1307.3150, 2013.
3. M. Blaum, J. Hafner, and S. Hetzler, Partial-MDS Codes and their Application to RAID Type of Architectures, CoRR, vol. abs/1205.0997, 2012.
4. J. S. Plank and M. Blaum, Sector-disk (SD) erasure codes for mixed failure modes in RAID systems, TOS, vol. 10, no. 1, p. 4, 2014.
5. M. Blaum, Construction of PMDS and SD codes extending RAID 5, CoRR, vol. abs/1305.0032, 2013.
6. M. Blaum and J. S. Plank, Construction of two SD codes, CoRR, vol. abs/1305.1221, 2013.
7. M. Blaum, J. S. Plank, M. Schwartz, and E. Yaakobi, Construction of partial MDS (PMDS) and sector-disk (SD) codes with two global parity symbols, CoRR, vol. abs/1401.4715, 2014
8. S. B. Balaji, P. V. Kumar, "On Partial Maximally-Recoverable and Maximally-Recoverable Codes", *CoRR*, abs/1501.07130, 2015.
9. V. Lalitha, Satyanarayana V. Lokam, "Weight Enumerators and Higher Support Weights of Maximally Recoverable Codes", abs/1507.01095, 2015.

Codes with Locality - Parallel Recovery

1. W. Song, S. H. Dau, C. Yuen, and T. Li, "Optimal locally repairable linear codes," *Selected Areas in Communications, IEEE Journal on*, 2014.
2. L. Pamies-Juarez, H. D. L. Hollmann, F. Oggier, "Locally repairable codes with multiple repair alternatives, *CoRR*, abs/1302.5518, 2013.
3. A. Wang and Z. Zhang, "Repair locality from a combinatorial perspective," ISIT 2014.
4. J.-H. Kim, M.-Y. Nam, and H.-Y. Song, "Binary locally repairable codes from complete multipartite graphs," in *ICTC, 2015*.
5. P. Huang, E. Yaakobi, H. Uchikawa, and P. H. Siegel, "Binary linear locally repairable codes," *CoRR*, vol. abs/1511.06960, 2015.
6. A. Wang, Z. Zhang, "Repair Locality with Multiple Erasure Tolerance," *CoRR,* abs/1306.4774, 2013.
7. Y.S. Rawat, D. Papailiopoulos, A.G. Dimakis, S. Vishwanath, "Locality and availability in distributed storage," ISIT 2014.
8. J. Zhang, X. Wang, and G. Ge, "Some improvements on locally repairable codes," *CoRR*, vol. abs/1506.04822, 2015.
9. I. Tamo, A. Barg, and A. Frolov, "Bounds on the parameters of locally recoverable codes," *CoRR*, vol. abs/1506.07196, 2015.
10. A. Wang, Z. Zhang, and M. Liu, "Achieving arbitrary locality and availability in binary codes," *ISIT*, 2015.
11. L. Shen, F. Fu, and X. Guang, "On the locality and availability of linear codes based on finite geometry," *IEICE Transactions*, 2015.

Codes with Locality - Sequential Recovery

1. N. Prakash, V. Lalitha, P. V. Kumar, "Codes with Locality for Two Erasures," ISIT 2014.
2. A. Rawat, A. Mazumdar, and S. Vishwanath, "On cooperative local repair in distributed storage," in *CISS*, 2014.
3. W. Song and C. Yuen, "Locally repairable codes with functional repair and multiple erasure tolerance," *CoRR*, vol. abs/1507.02796, 2015.
4. W. Song and C. Yuen, "Binary locally repairable codes - sequential repair for multiple erasures," *CoRR*, vol. abs/1511.06034, 2015.
5. S. B. Balaji, K. P. Prasanth, and P. V. Kumar, "Binary codes with locality for multiple erasures having short block length," *CoRR*, abs/1601.07122, 2016.

Additional Papers on Codes with Locality I

1. V. R. Cadambe and A. Mazumdar, "Bounds on the Size of Locally Recoverable Codes," in IEEE Trans. on Information Theory, vol. 61, no. 11, pp. 5787-5794, Nov. 2015.
2. T. Westerbäck, R. Freij, T. Ernvall and C. Hollanti, "On the Combinatorics of Locally Repairable Codes via Matroid Theory", *CoRR*, abs/1501.00153, 2015.
3. I. Tamo, D. S. Papailiopoulos, and A. G. Dimakis, Optimal locally repairable codes and connections to matroid theory, ISIT 2013.
4. N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, Optimal locally repairable codes via rank-metric codes, ISIT 2013.
5. A. Wang and Z. Zhang, "An Integer Programming Based Bound for Locally Repairable Codes", *CoRR*, abs/1409.0952, 2014.
6. S. Goparaju and R. Calderbank, "Binary cyclic codes that are locally repairable," ISIT 2014
7. T. Westerbck, T. Ernvall and C. Hollanti, "Almost affine locally repairable codes and matroid theory," ITW 2014.
8. H. Xie and Z. Yan, "Two-layer locally repairable codes for distributed storage systems," ICC 2014.
9. T. Ernvall, T. Westerbck and C. Hollanti, "Constructions of optimal and almost optimal locally repairable codes," (VITAE), 2014 .
10. T. Ernvall, T. Westerbäck and C. Hollanti, "Linear Locally Repairable Codes with Random Matrices",*CoRR*, abs/1408.0180, 2014.

Additional Papers on Codes with Locality II

11. B. Sasidharan, G. K. Agarwal and P. V. Kumar, "Codes with hierarchical locality," Information Theory (ISIT), ISIT 2015.
12. H. M. Kiah, S. H. Dau, W. Song, C. Yuen, "Local Codes with Addition Based Repair", abs/1506.02349, 2015.
13. T. Westerbäck, R. Freij-Hollanti and C. Hollanti, "Applications of Polymatroid Theory to Distributed Storage Systems", *CoRR*, abs/1510.02499, 2015.
14. N. Silberstein, A. Zeh, "Optimal Binary Locally Repairable Codes via Anticodes", abs/1501.07114, 2015.
15. I. Tamo, A. Barg, S. Goparaju and R. Calderbank, "Cyclic LRC Codes, binary LRC codes, and upper bounds on the distance of cyclic codes", *CoRR*, abs/1603.08878, 2016.
16. A. Barg, I. Tamo, S. Vladuts, "Locally recoverable codes on algebraic curves", *CoRR*, abs/1603.08876, 2016.
17. T. Ernvall, T. Westerbck, R. Freij-Hollanti, C. Hollanti, "A Connection Between Locally Repairable Codes and Exact Regenerating Codes" , abs/1603.05846, 2016.
18. A. Pöllänen, T. Westerbäck, R. Freij-Hollanti and C. Hollanti, "Improved Singleton-type Bounds for Locally Repairable Codes", *CoRR*, abs/1602.04482, 2016.
19. J. Hao and Shu-Tao Xia, "Bounds and Constructions of Locally Repairable Codes: Parity-check Matrix Approach", *CoRR*, abs/1601.05595, 2016.
20. A. Zeh and E. Yaakobi, "Bounds and Constructions of Codes with Multiple Localities", *CoRR*, abs/1601.02763, 2016.

Additional Papers on Codes with Locality III

21. A.Zeh and E. Yaakobi, "Optimal Linear and Cyclic Locally Repairable Codes over Small Fields", *CoRR*, abs/1502.06809, 2015.
22. S. Kadhe and A. Sprintson, "Codes with Unequal Locality", *CoRR*, abs/1601.06153, 2016.
23. T. Ernvall, T. Westerbck, R. Freij-Hollanti and C. Hollanti, "Constructions and Properties of Linear Locally Repairable Codes," in IEEE Trans. on Information Theory, 2016.
24. M. Blaum, S. R. Hetzler, "Integrated Interleaved Codes as Locally Recoverable Codes: Properties and Performance", abs/1602.02704, 2016.
25. R. Freij-Hollanti, T. Westerbck and C. Hollanti, "Locally Repairable Codes with Availability and Hierarchy: Matroid Theory via Examples", (IZS) 2016.
26. Siddhartha Kumar, Alexandre Graell i Amat, Iryna Andriyanova, Fredrik Brännström, "A Family of Erasure Correcting Codes with Low Repair Bandwidth and Low Repair Complexity," *IEEE Global Communications Conference (GLOBECOM) (2015)*.

Batch Codes

1. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," *36th Annual ACM Symposium on Theory of Computing,* New York, 2004
2. N. Silberstein and A. Gal, "Optimal combinatorial batch codes based on block designs," *Designs, Codes and Cryptography,* 2014

Papers Discussing System Employing Codes for Distributed Storage

1. K. V. Rashmi, N. B. Shah, D. Gu, H. Kuang, D. Borthakur and K. Ramchandran, "A hitchhiker's guide to fast and efficient data reconstruction in erasure-coded data centers", *SIGCOMM,* 2014
2. M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, D. Borthakur, "Xoring elephants: Novel erasure codes for big data," 2013.
3. A. Duminuco, E. Biersack, "A practical study of regenerating codes for peer-to-peer backup systems," *IEEE International Conference on Distributed Computing Systems,* 2009.
4. C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li and S. Yekhanin, "Erasure coding in Windows Azure Storage", in *USENIX Annual Technical Conference (ATC)*, 2012
5. O. Khan, R. Burns, J. S. Plank, W. Pierce and C. Huang. "Rethinking Erasure Codes for Cloud File Systems: Minimizing I/O for Recovery and Degraded Reads," *FAST 2012: 10th USENIX Conference on File and Storage Technologies,* 2012
6. M. N. Krishnan, N. Prakash, V. Lalitha, B. Sasidharan, P. V, Kumar, S. Narayanamurthy, R. Kumar and S. Nandi, "Evaluation of codes with inherent double replication for Hadoop", in *Proc. USENIX HotStorage,* 2014
7. M. Xia, M. Saxena, M. Blaum and D. Pease, "A Tale of Two Erasure Codes in HDFS", *FAST,* 2015
8. K. V. Rashmi, P. Nakkiran, J. Wang, N. B. Shah and K. Ramchandran, "Having Your Cake and Eating It Too: Jointly Optimal Erasure Codes for I/O, Storage, and Network-bandwidth",*FAST,* 2015

Thanks!