Recent Developments in Coding for Distributed Storage

P. Vijay Kumar

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

Adjunct Professor, EE-Systems
University of Southern California

Workshop on Emerging Wireless Networks
Institute for Pure and Applied Mathematics
University of California, Los Angeles

Feb. 6-10, 2017

Thanks to Christina Fragouli and Suhas Diggavi and for the kind invite....

Collaborators

Past and Present Research Collaborators

- Alexander Barg, Ithak Tamo
- S. Narayanamurthy, R. Kumar, S. Husein and Siddhartha Nandi (NetApp Inc., India)

Past and Present Students

- Narayanamoorthy Prakash (Post-Doc, MIT)
- Lalitha Vadlamani (Faculty, IIITH)

- Gaurav Agrawal (now pursuing PhD at UCLA)
- K. P. Prasanth (Qualcomm, India)

- Birenjith Sasidharan (PhD student, to graduate shortly)
- S. B. Balaji, Nikhil M. Krishnan, Myna Vajha (PhD students)
- Ganesh Kini, Bhagyashree Puranik, R. Vinayak (Master's students)

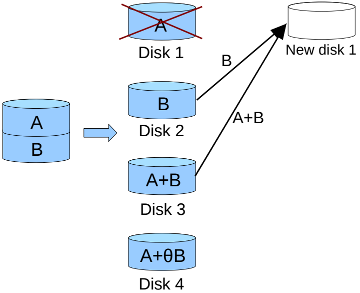
Outline

- Codes with Local and Sequential Erasure Recovery
- Improving Decoding Performance Through Locality
- Codes with Hierarchical Locality
- A Recent High-Rate Regenerating Code Construction (briefly)

Inefficiency of Linear Node Repair in an MDS Code

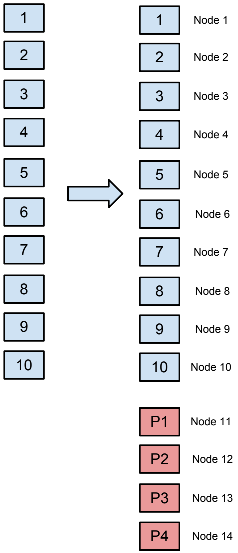
An obvious approach:

- Connect to any 2 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!

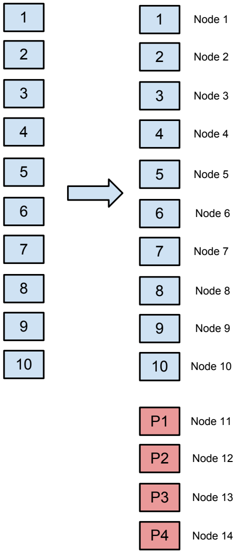
A Second Example: Facebook's Code



- $[14, 10]$ MDS code
- Has the "any 10 out of 14" property
- Used in Facebook data centers

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

The Facebook Code is Inefficient in Number of Helper Nodes Needed

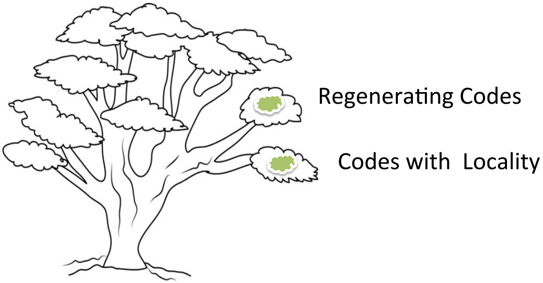


- Needs to connect to 10 nodes to repair a failed node
- This calls for interrupting operations in 10 nodes (apart from downloading the entire data file)

- 10 is the *repair degree*

- Are there better options ?

Adding a Branch (or two) to Coding Theory



Regenerating Codes

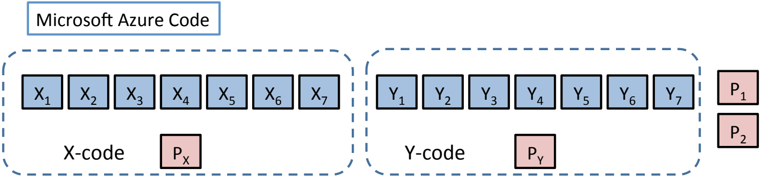
Codes with Locality

- Regenerating codes reduce repair bandwidth
- Codes with locality reduce repair degree

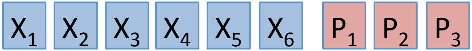
- A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Trans. Inform. Th.*, Sep. 2010.
- P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the Locality of Codeword Symbols," *IEEE Trans. Inf. Theory*, Nov. 2012.

Image: `http://www.colorluna.com`

Codes with Locality to Minimize Reduce Repair Degree (Example of Windows Azure Storage)



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.

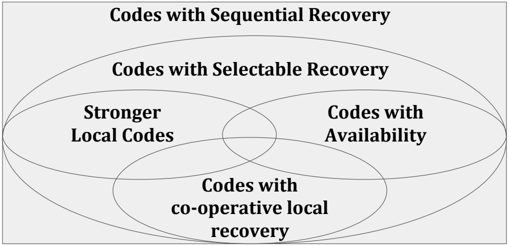


Huang, Simitci, Xu, Ogus, Calder, Gopalan, Li, Yekhanin, "Erasure Coding in Windows Azure Storage," USENIX, Boston, MA, 2012.

Codes with Locality For Multiple Erasures

Some approaches:

- **Sequential Approach** - Recovery from at most t erasures. There exists at least one sequence of erased code symbol recovery at every step.
- **Availability** - For every code symbol there are multiple mutually disjoint recovery sets.
- **Selectable Recovery** - Every erased symbol has a parity check that does not involve any other erased symbol.


Codes with Sequential Recovery
Codes with Selectable Recovery
Stronger Local Codes
Codes with Availability
Codes with co-operative local recovery

Codes with Locality for Sequential Recovery

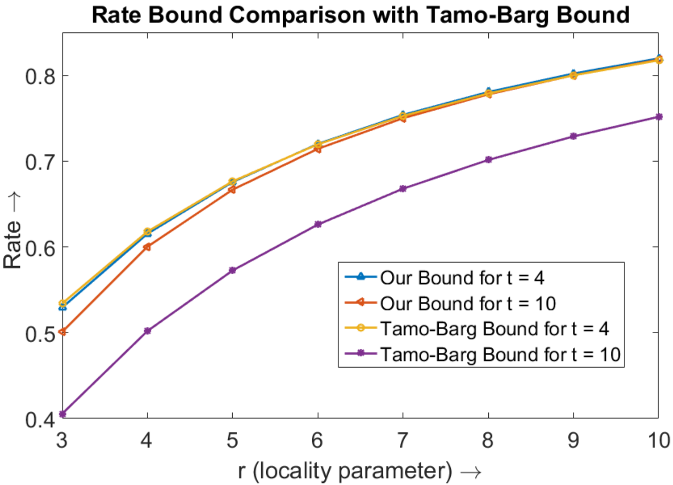
Definition

An $[n, k]$ code over a field $\mathbb{F}_q$ is defined as a **code with sequential recovery** from t erasures having locality r if for any set of $s \leq t$ erased symbols, $\{c_{\sigma_1}, ..., c_{\sigma_s}\}$, there exists a codeword $\underline{h}$ in the dual of the code, of Hamming weight $\leq r+1$, such that $\text{supp}(\underline{h}) \cap \{\sigma_1, ..., \sigma_s\} = 1$.

We denote the above defined codes as $(n, k, r, t)_{seq}$ codes.

N. Prakash, V. Lalitha, P. Vijay Kumar, "Codes with Locality for Two Erasures," ISIT 2014.
S. B. Balaji, Ganesh R. Kini and PVK, "A Tight Rate Bound and a Matching Construction for Locally Recoverable Codes with Sequential Recovery From Any Number of Multiple Erasures," arXiv:1611.08561v6 [cs.IT] 9 Feb 2017.

Comparison of Sequential Recovery and Availability



Upper Bound on Rate Under Sequential, Local Recovery

Theorem

Rate Bound: *Let $\mathcal{C}$ be an $(n, k, r, t)_{seq}$ code over a field $\mathbb{F}_q$. Let $r \geq 3$. Then*

$$\frac{k}{n} \leq \frac{r^{\frac{t}{2}}}{r^{\frac{t}{2}} + 2\sum_{i=0}^{\frac{t}{2}-1} r^i} \quad \text{for } t \text{ an even integer,} \tag{1}$$

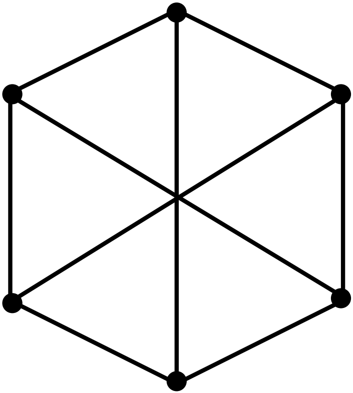
$$\frac{k}{n} \leq \frac{r^s}{r^s + 2\sum_{i=1}^{s-1} r^i + 1} \quad \text{for } t \text{ an odd integer,} \tag{2}$$

where $s = \frac{t+1}{2}$.

Proof By Inferring Form of Parity-Check Matrix

$$H = \left[\begin{array}{c|c|c|c|c|c|c|c|c}
D_0 & A_1 & 0 & 0 & \dots & 0 & 0 & 0 & \\ \cline{1-8}
0 & D_1 & A_2 & 0 & \dots & 0 & 0 & 0 & \\ \cline{1-8}
0 & 0 & D_2 & A_3 & \dots & 0 & 0 & 0 & \\ \cline{1-8}
0 & 0 & 0 & D_3 & \dots & 0 & 0 & 0 & \\ \cline{1-8}
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & D \\ \cline{1-8}
0 & 0 & 0 & 0 & \dots & A_{\frac{t}{2}-2} & 0 & 0 & \\ \cline{1-8}
0 & 0 & 0 & 0 & \dots & D_{\frac{t}{2}-2} & A_{\frac{t}{2}-1} & 0 & \\ \cline{1-8}
0 & 0 & 0 & 0 & \dots & 0 & D_{\frac{t}{2}-1} & & \\ \cline{1-7}
0 & 0 & 0 & 0 & \dots & 0 & 0 & C &
\end{array}\right]$$

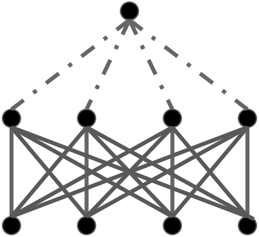
Regular Graph Construction ($t = 2, r = 3$)



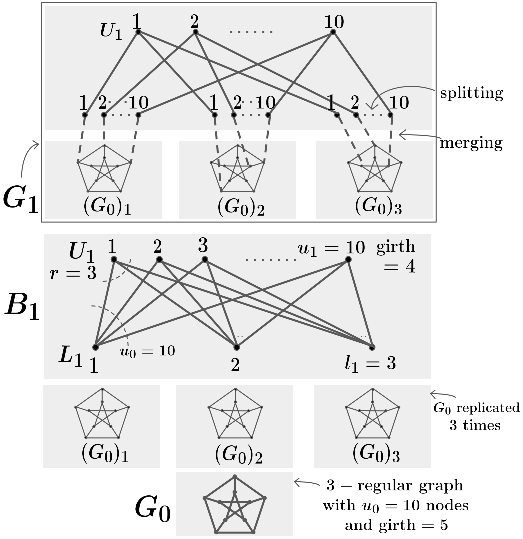
- Edges correspond to symbols
- Nodes correspond to parity-checks

Product Code for ($t = 3$, $r = 4$)

c_{11}	c_{12}	c_{13}	c_{14}	c_{15}
c_{21}	c_{22}	c_{23}	c_{24}	c_{25}
c_{31}	c_{32}	c_{33}	c_{34}	c_{35}
c_{41}	c_{42}	c_{43}	c_{44}	c_{45}
c_{51}	c_{52}	c_{53}	c_{54}	c_{55}



A Binary Rate-Optimal Code ($t = 4, r = 3$)



Locality and Decoding (BCH Codes)

Desired Null Spectrum of the BCH Code (zeros of the generator polynomial)

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118
119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203
204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237
238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254

▨ distance zeros

Desired Null Spectrum + Conjugacy

▣ 0	▣ 1	▣ 2	▣ 3	▣ 4	▣ 5	▣ 6	▣ 7	▣ 8	▣ 9	▣ 10	▣ 11	▣ 12	13	▨ 14	15	▨ 16
17	▨ 18	19	▨ 20	21	▨ 22	23	▨ 24	25	26	27	▨ 28	29	30	31	▨ 32	▨ 33
34	35	▨ 36	37	38	39	▨ 40	41	42	43	▨ 44	45	46	47	▨ 48	49	50
51	52	53	54	55	▨ 56	57	58	59	60	61	62	63	▨ 64	▨ 65	▨ 66	67
68	69	70	71	▨ 72	73	74	75	76	77	78	79	▨ 80	81	82	83	84
85	86	87	▨ 88	89	90	91	92	93	94	95	▨ 96	▨ 97	98	99	100	101
102	103	104	105	106	107	108	109	110	111	▨ 112	113	114	115	116	117	118
119	120	121	122	123	124	125	126	127	▨ 128	▨ 129	▨ 130	▨ 131	▨ 132	▨ 133	134	135
136	137	138	139	140	141	142	143	▨ 144	145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	▨ 160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	▨ 176	177	178	179	180	181	182	183	184	185	186
187	188	189	190	191	▨ 192	▨ 193	▨ 194	195	196	197	198	199	200	201	202	203
204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	▨ 224	225	226	227	228	229	230	231	232	233	234	235	236	237
238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254

▣ distance zeros ▨ distance zeros (cyc. cosets)

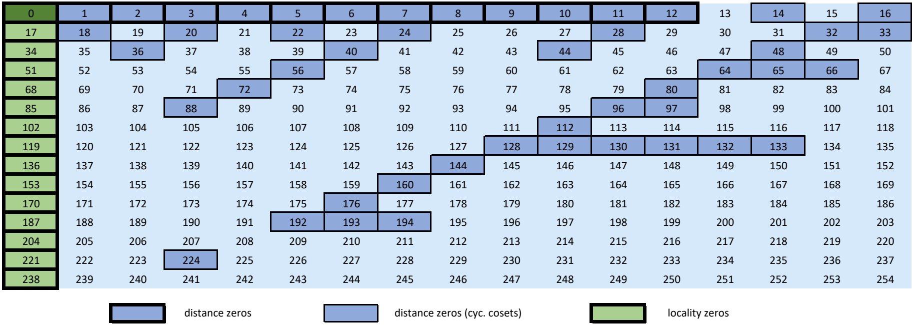
Desired Null Spectrum for Locality = Locality Plus Conjugacy

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118
119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203
204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237
238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254

🟩 locality zeros

- Itzhak Tamo, Alexander Barg, Sreechakra Goparaju, Robert Calderbank, “Cyclic LRC Codes, binary LRC codes, and upper bounds on the distance of cyclic codes,” ISIT 2015.

Final Null Spectrum



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118
119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169
170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203
204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220
221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237
238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254

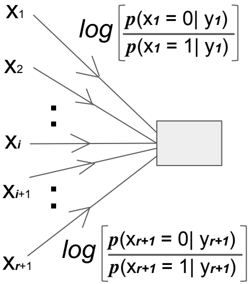
▣ distance zeros

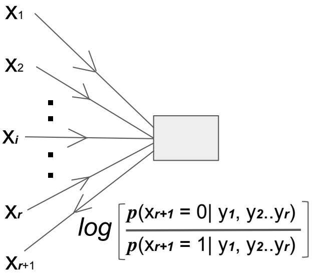
🟦 distance zeros (cyc. cosets)

🟩 locality zeros

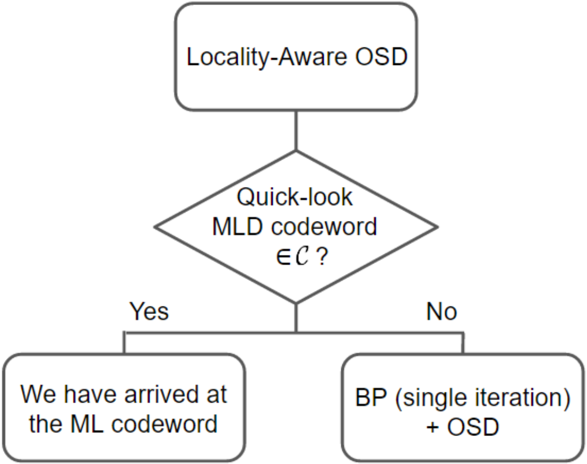
Belief Propagation (BP+OSD) of a BCH Code

Assumption: Improve beliefs through one round of belief propagation.


x_1
x_2
x_i
x_{i+1}
x_{r+1}
$log\left[\frac{p(x_1 = 0|\ y_1)}{p(x_1 = 1|\ y_1)}\right]$
$log\left[\frac{p(x_{r+1} = 0|\ y_{r+1})}{p(x_{r+1} = 1|\ y_{r+1})}\right]$


x_1
x_2
x_i
x_r
x_{r+1}
$log\left[\frac{p(x_{r+1} = 0|\ y_1, y_2..y_r)}{p(x_{r+1} = 1|\ y_1, y_2..y_r)}\right]$

Quick-Look ML Decoder



Locality-Aware OSD

Quick-look MLD codeword $\in \mathcal{C}$?

Yes

No

We have arrived at the ML codeword

BP (single iteration) + OSD

Ordered Statistics Decoding

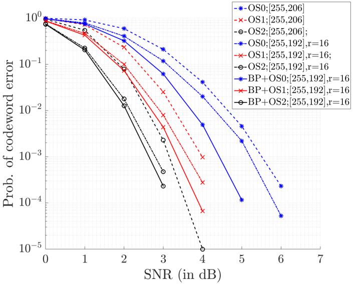
- Hard decision decoding based on most reliable bits (Order-0)
  - Sort the symbols of the received vector in descending order of reliability.
  - Hard decision decode the first k independent bits (Most Reliable Independent, MRI) bits and obtain the n bit codeword.
- Order-l reprocessing
  - For $0 \leq i \leq l$, change each of the i bits of the k MRI bits to obtain $m = \sum_{i=0}^{l} \binom{k}{i}$ message vectors.
  - Find the m codewords corresponding to the m information vectors.
  - Pick the codeword closest to the received vector.

- Marc P. C. Fossorier and Shu Lin "Soft-Decision Decoding of Linear Block Codes Based on Ordered Statistics," IT-Trans, Sep. 1995.

Performance of Locality-aware OSD

We consider two codes in binary input AWGN channel:

- $\mathcal{C}_1$: $[255, 206]$ BCH code $\rightarrow$ doesn't have any locality.
- $\mathcal{C}_2$: $[255, 192]$ BCH-like $\rightarrow$ constructed by adding locality $r = 16$ in $\mathcal{C}_1$.



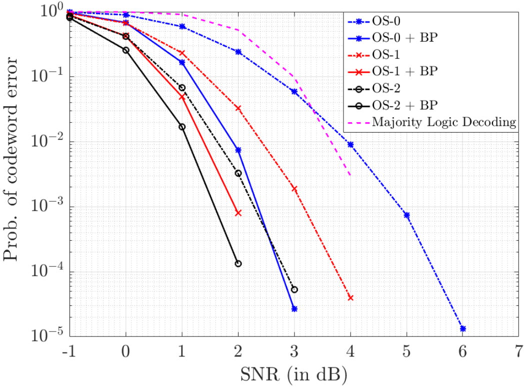
M. Nikhil Krishnan, Bhagyashree Puranik, PVK, Itzhak Tamo, and Alexander Barg, "A Study on the Impact of Locality in the Decoding of Binary Cyclic Codes, to be uploaded to arXiv.

Scheme-1: Locality-aware Ordered Statistics Decoding

- In the next slide, we consider decoding a type-1 Doubly Transitive Invariant Code that is majority logic decodable, with parameters $[255, 175]$, having availability, $t = 16$ and locality, $r = 15$.

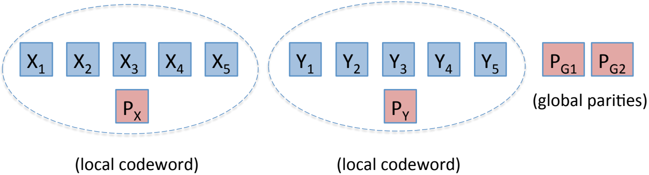
Performance of Locality-Aware OSD on an Availability Code

- Comparison of OSD, BP+OSD, and majority logic decoding of a $[255, 175]$ type-I DTI code, locality $r = 15$, availability $t = 16$ in binary input AWGN channel.

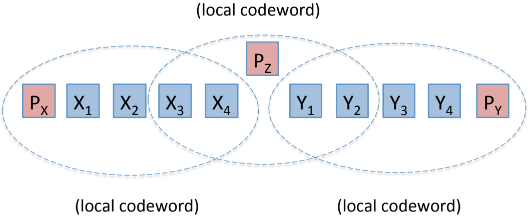


Hierarchical Locality

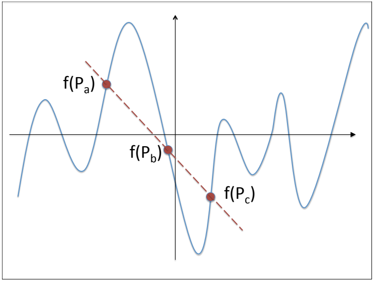
Windows Azure Uses Information Locality, How About All-Symbol Locality ?



(information-symbol locality)



The Tamo-Barg Approach to All-Symbol Locality)



- select subset of polynomials : $(f(P_1), \cdots, f(P_n))$, with $\deg(f) \leq (k-1)$
- such that: given point P_a there exist other points fitted by a lower-degree polynomial
- such as a line, which can be used for correction
- There is also a Chinese Remainder Theorem interpretation

Itzhak Tamo, Alexander Barg, "A family of optimal locally recoverable codes, " *IEEE Transactions on Information Theory*. Aug 2014.

Chinese Remainder Theorem

Sun Zi Suanjing *Master Sun's Mathematical Manual* Problem 26, Volume 3 (estimated to be published 300-500 AD) reads: "There are certain things whose number is unknown. A number is repeatedly divided by 3, the remainder is 2; divided by 5, the remainder is 3; and by 7, the remainder is 2. What will the number be ?" The problem can be expressed as

$$x \equiv 2 \pmod 3 \equiv 3 \pmod 5 \equiv 2 \pmod 7.$$

Sun Zi solved the problem as we do,

(mod 3)	(mod 5)	(mod 7)	$x = ?$
1	0	0	70
0	1	0	21
0	0	1	15

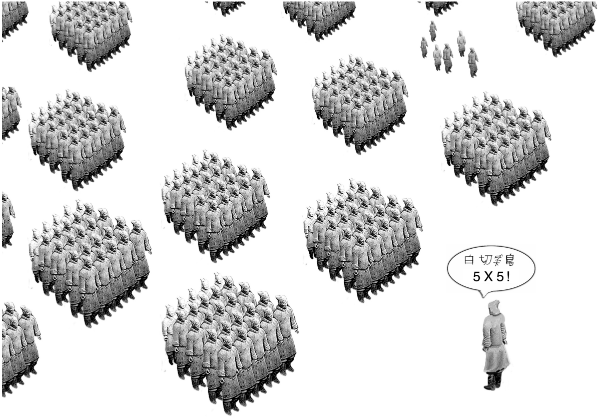
giving

$$\mathbf{x} = 2(70) + 3(21) + 2(15) \equiv 233 \equiv \mathbf{23} \pmod{105}.$$

- Shen Kangsheng, *Historical Development of the Chinese Remainder Theorem*.
- Wikipedia, *Chinese Remainder Theorem* .

Chinese Remainder Theorem

One interpretation of the motivation[1]



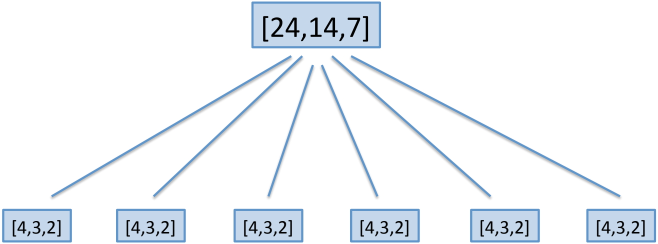
$N \equiv 6 \pmod{25}$

[1] Figure from "A Mechanical Proof of the Chinese Remainder Theorem," David M. Russinoff, Advanced Micro Devices. . Advanced Micro Devices, Inc. `http://russinoff.com/papers/soldiers.jpeg`.

Codes with Hierarchical Locality

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

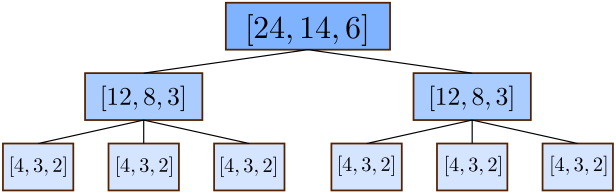
Codes with Locality do not Scale



- If the local code is overwhelmed, then one has to appeal to the overall code which means contacting all 14 nodes for node repair.
- So how does one ensure scalability [2] ?

[2] Question posed during University Melbourne talk.

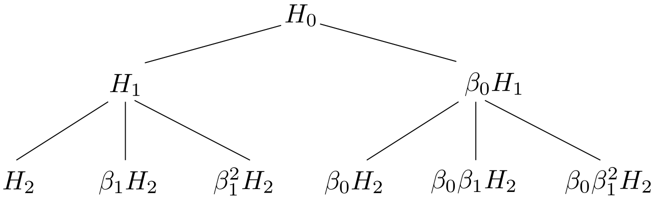
Codes with Hierarchical Locality



- Codes with hierarchical locality ensure scalability by providing an intermediate layer of codes
- these can help when the when the local code at the bottom, fails.

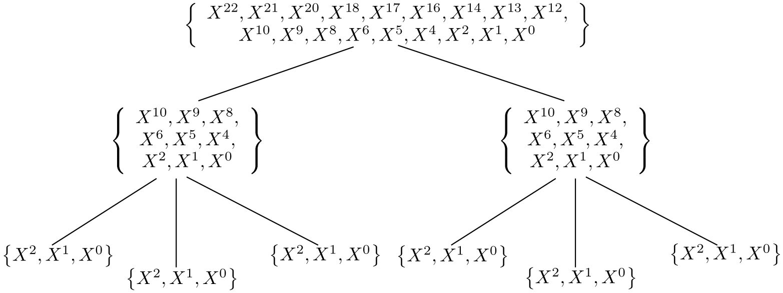
Hierarchical Construction Uses Groups and Subgroups to Partition Coordinates

- Code lengths must satisfy divisibility condition $n_2 \mid n_1 \mid n$
- Here $4 \mid 12 \mid 24$.



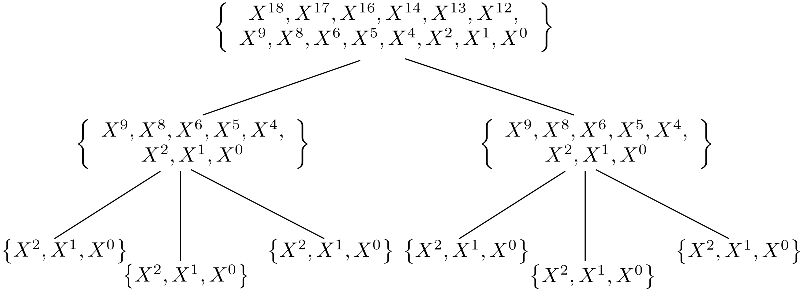
1. Subgroup chain $H_2 \subseteq H_1 \subseteq H_0$
2. With sizes given by $4, 12, 24$

Chinese-Remainder-Theorem-View of Hierarchical Codes


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

- Impact in higher layers of
  - restricting polynomials in bottom layer to have degree 2, rather than the customary 3.

A Final Refinement for Strengthening Middle and Top-Level Codes



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

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

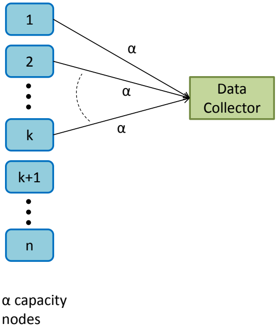
$\{X^2, X^1, X^0\}$ $\{X^2, X^1, X^0\}$ $\{X^2, X^1, X^0\}$ $\{X^2, X^1, X^0\}$ $\{X^2, X^1, X^0\}$ $\{X^2, X^1, X^0\}$

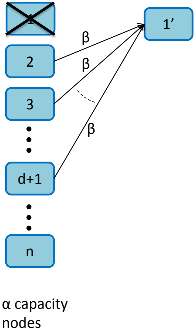
- Restricting degree of polynomials in middle and top layers strengthens the corresponding codes.

Regenerating Codes

Regenerating Codes - Formal Definition

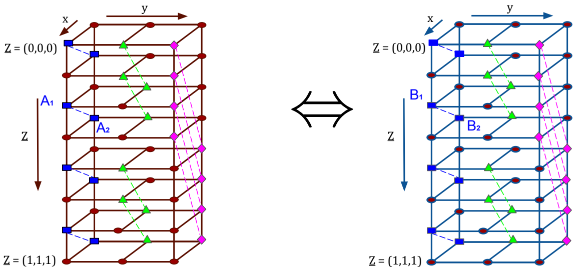
Parameters: $(\ (n, k, d), \ (\alpha, \beta), \ B, \ \mathbb{F}_q \ )$





- 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)

High-Rate MSR Code



- Min Ye, Alexander Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," arXiv:1605.08630v1, 27 May 2016.
- Birenjith Sasidharan, Myna Vajha, P. Vijay Kumar, "An Explicit, Coupled-Layer Construction of a High-Rate MSR Code with Low Sub-Packetization Level, Small Field Size and $d < (n-1)$," arXiv:1701.07447v1, 25 Jan 2017.

Thanks!