Codes for Distributed Storage: Theory and Practice

Myna Vajha
Department of ECE, IISc Bangalore.

STCS Symposium, TIFR

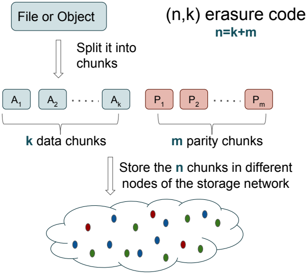
1st March, 2020

Collaborators

- My advisor Prof. Vijay Kumar
- Birenjith Sasidharan, Balaji S. B (MSR constructions)
- IISc: Vinayak Ramkumar, Bhagyasree Puranik and Ganesh Kini (systems evaluation)
- Netapp: Srinivas Narayanamurthy, Syed Hussain, Siddhartha Nandi (systems evaluation)
- University of Maryland: Min Ye and Alexandar Barg (systems evaluation)

Erasure Coding for Fault Tolerance

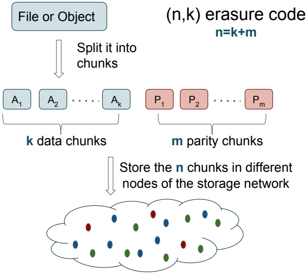
- Fault tolerance is achieved using erasure coding



The n chunks taken together, form a stripe.

Erasure Coding for Fault Tolerance

- Fault tolerance is achieved using erasure coding



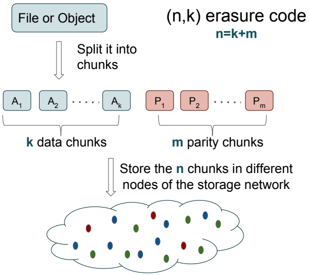
The n chunks taken together, form a stripe.

Two Key Performance Measures

1. Storage Overhead $\frac{n}{k}$
2. Fault Tolerance - at most m storage units

Erasure Coding for Fault Tolerance

- Fault tolerance is achieved using erasure coding



The n chunks taken together, form a stripe.

Two Key Performance Measures

1. Storage Overhead $\frac{n}{k}$
2. Fault Tolerance - at most m storage units

MDS Codes

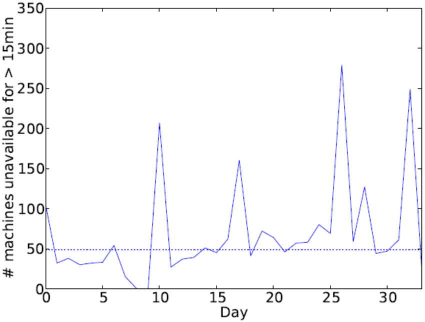
1. For given (n, k), **MDS erasure codes** have the maximum-possible fault tolerance
2. RAID 6 and Reed-Solomon codes are examples of MDS codes.

RS Codes in Practice

Storage Systems	Reed-Solomon codes
IBM Spectrum Scale RAID	RS(10,8) and RS(11,8)
Linux RAID-6	RS(10,8)
Google File System II (Colossus)	RS(9,6)
Quantcast File System	RS(9,6)
Hadoop Distributed File System 3	RS(9,6)
Yahoo Cloud Object Store	RS(11,8)
Backblaze's online backup	RS(20,17)
Facebook's f4 BLOB storage system	RS(14,10)
Baidu's Atlas Cloud Storage	RS(12, 8)

H. Dau et al, "Repairing Reed-Solomon Codes with Single and Multiple Erasures," ITA, 2017, San Diego.

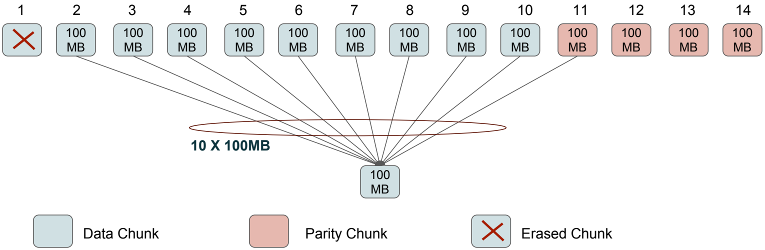
Erasure Codes and Node Failures



- A median of 50 nodes are unavailable per day.
- 98% of the failures are single chunk failures.
- A median of 180TB of network traffic per day is generated in order to reconstruct the RS coded data corresponding to unavailable machines.
- **Thus there is a strong need for erasure codes that can efficiently recover from single-node failures.**

Image courtesy: Rashmi et al.: "A Solution to the Network Challenges of Data Recovery in Erasure-coded Distributed Storage Systems: A Study on the Facebook Warehouse Cluster," USENIX Hotstorage, 2013.

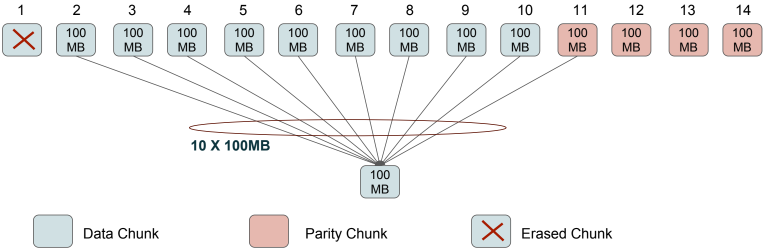
Conventional Node Repair of an RS Code



In the example $(14, 10)$ RS code,

1. the amount of data downloaded to repair 100MB of data equals 1GB.

Conventional Node Repair of an RS Code



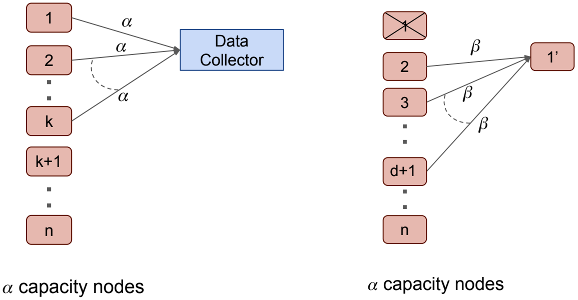
In the example $(14, 10)$ RS code,

1. the amount of data downloaded to repair 100MB of data equals 1GB.

clearly, there is room for improvement...

Regenerating Codes

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



- Data (of size B) can be recovered by connecting to any k of n nodes
- A failed node can be repaired by connecting to any d nodes, downloading β symbols from each node; ($d\beta <<$ file size B)

Dimakis et al. Network Coding for Distributed Storage Systems

Regenerating Codes

1. Optimal File size B possible by an (n, k, d, α, β) regenerating code:

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

Regenerating Codes

1. Optimal File size B possible by an (n, k, d, α, β) regenerating code:

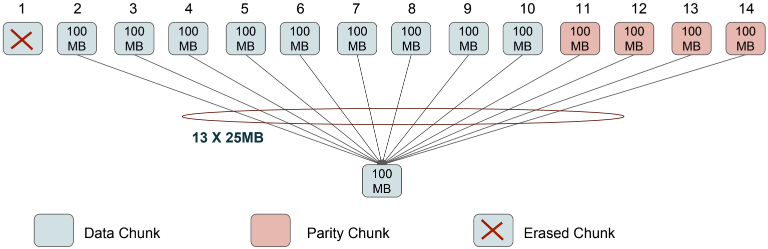
$$
\begin{aligned}
B &= \sum_{i=0}^{k-1} \min(\alpha, (d-i)\beta) \\
&\leq k\alpha
\end{aligned}
$$

2. Minimum storage regenerating (MSR) codes are a subclass of regenerating codes such that:

$$
\alpha = \frac{B}{k}, \quad \beta = \frac{\alpha}{d-k+1}
$$

3. We restrict to Minimum-Storage-Regenerating (MSR) codes – repair-optimal MDS codes.

MSR Codes

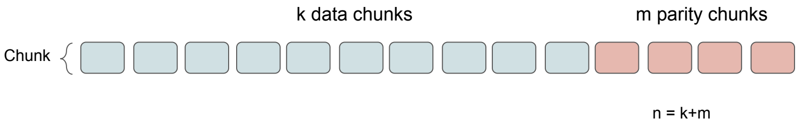


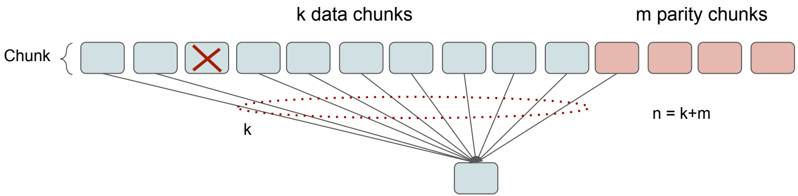
1. Size of failed node's contents: 100MB
2. RS repair BW: 1 GB
3. MSR Repair BW: 325 MB

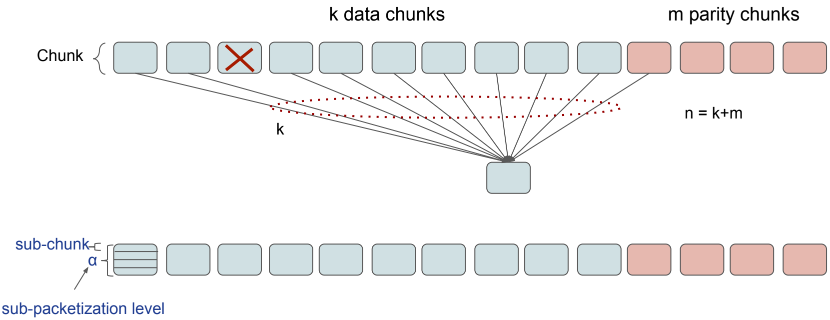
Key to the Impressive, Low-Repair BW of MSR Codes

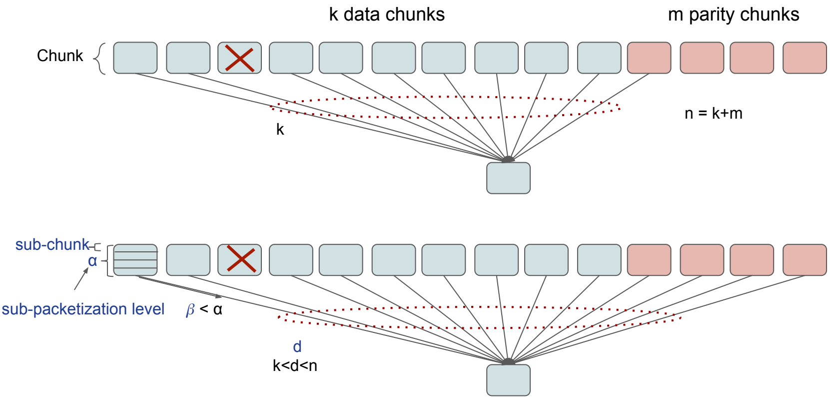
Key to the Impressive, Low-Repair BW of MSR Codes

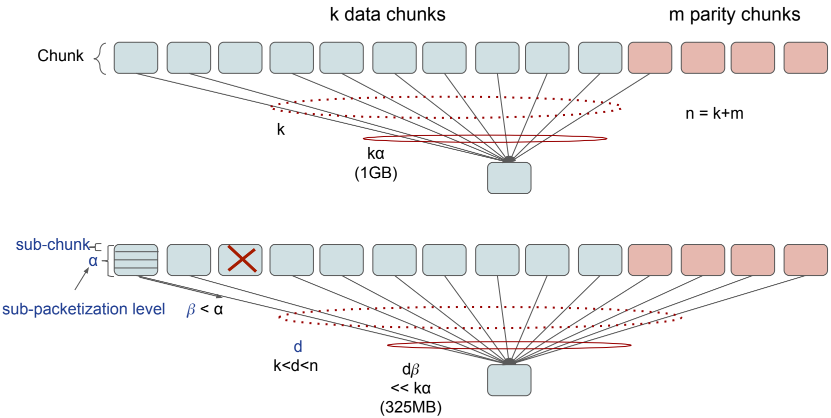
In a nutshell: sub-packetization... we explain...

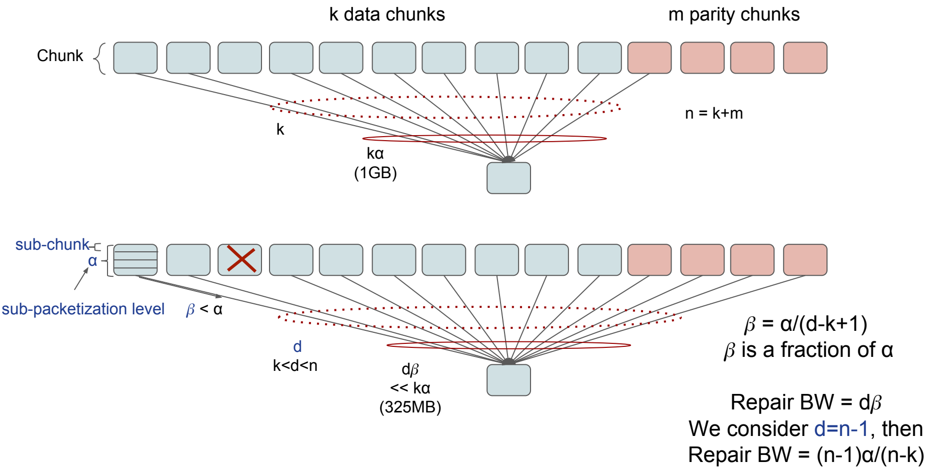

k data chunks
m parity chunks
Chunk
n = k+m


k data chunks
m parity chunks
Chunk
k
n = k+m

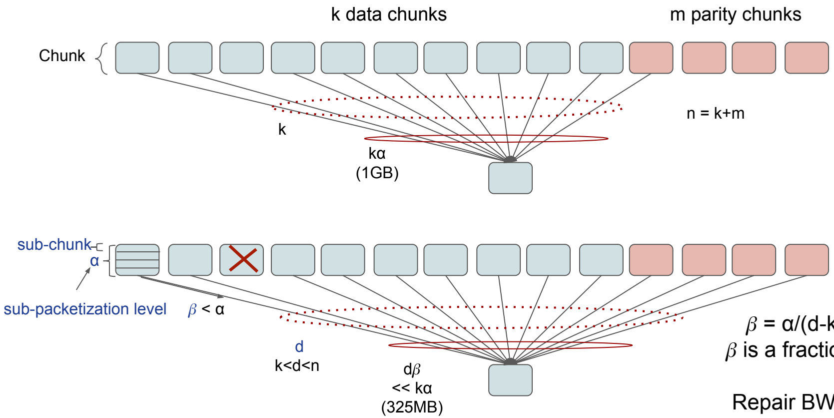

k data chunks
m parity chunks
Chunk
k
n = k+m
sub-chunk
α
sub-packetization level


k data chunks
m parity chunks
Chunk
k
n = k+m
sub-chunk
α
sub-packetization level
β < α
d
k<d<n


k data chunks
m parity chunks
Chunk
n = k+m
k
kα
(1GB)
sub-chunk
α
sub-packetization level
β < α
d
k<d<n
dβ
<< kα
(325MB)



k data chunks
m parity chunks
Chunk
n = k+m
k
kα
(1GB)
sub-chunk
α
sub-packetization level
β < α
d
k<d<n
dβ
<< kα
(325MB)
β = α/(d-k+1)
β is a fraction of α
Repair BW = dβ
We consider d=n-1, then
Repair BW = (n-1)α/(n-k)



$\beta = \alpha/(d-k+1)$

β is a fraction of α

Repair BW = dβ

We consider d=n-1, then

Repair BW = (n-1)α/(n-k)

Larger the m=n-k, larger the savings!!

Additional Properties Desired of an MSR Code

1. Minimal Disk Read (Optimal Access): Read exactly what is needed to be transferred
2. Minimize sub-packetization level α
3. Small field size, low-complexity implementation.
4. Two family of constructions
   - Coupled Layer (CLay) MSR code ($d = n - 1$)
   - Small d MSR code ($d = k + 1, k + 2, k + 3$)

4-way Optimality of Clay code

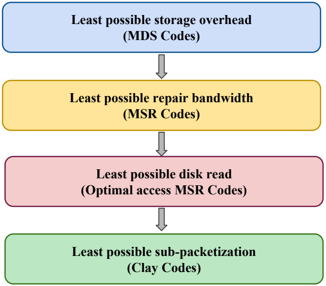


Image courtesy: denverpost.com

Putting Clay codes in perspective

- Given (n, k, d) let $s = d - k + 1$, $r = n - k$

Putting Clay codes in perspective

- Given (n, k, d) let $s = d - k + 1$, $r = n - k$

MSR Code	Parameters	α	Field Size (q)	All Node Repair	Optimal Access
Shah et al.	$(n, k, d = n-1 \geq 2k-1)$	r	$2r$	No	Yes
Suh et al.	$(n, k, d \geq 2k-1)$ $(n, k \leq 3, d)$	s	$2r$	Yes	No
Rashmi et al.	$(n \geq 2k-1, k, d)$	r	n	Yes	No
Papailiopoulos et al.	$(n, k, d = n-1)$	r^k	non-explicit	No	No
Tamo et al. Wang et al.	$(n, k, d = n-1)$	r^{k+1}	≤ 4 when $r \leq 3$, else non-explicit	Yes	Yes
Cadambe et al.	$(n \geq \frac{3k}{2}, k, d = n-1)$	$O(k^2)$	non-explicit	No	Yes
Sasidharan et al.	$(n, k, d = n-1)$	$r^{\lceil \frac{n}{r} \rceil}$	$O(n^r)$	Yes	Yes
Goparaju et al.	(n, k, d)	$s^{k\binom{r}{s}}$	-	No	Yes
Rawat et al.	(n, k, d)	$s^{\lceil \frac{n}{s} \rceil}$	$O(n^r)$	Yes	Yes
Ye & Barg (1a)	(n, k, d)	s^n	sn	Yes	No
Ye & Barg (1b)	(n, k, d)	s^{n-1}	$n+1$	Yes	Yes

Literature on High-Rate, OA MSR Codes with Optimum α

$$(n, k, d = n-1, \alpha = r^{\lceil \frac{n}{r} \rceil}), \ q \geq r\lceil \frac{n}{r} \rceil$$

- In May 2016, Ye & Barg came up with MSR codes that are based on Vandermonde RS codes.
- In July 2016, Sasidharan et.al came up with MSR codes that could be constructed from any MDS code.
- In ISIT 2017, Li et.al came up with a transformation that could convert any scalar MDS code to MSR construction.

Literature on High-Rate, OA MSR Codes with Optimum α

$$(n, k, d = n - 1, \alpha = r^{\lceil \frac{n}{r} \rceil}), \ q \geq r\lceil \frac{n}{r} \rceil$$

- In May 2016, Ye & Barg came up with MSR codes that are based on Vandermonde RS codes.
- In July 2016, Sasidharan et.al came up with MSR codes that could be constructed from any MDS code.
- In ISIT 2017, Li et.al came up with a transformation that could convert any scalar MDS code to MSR construction.
- In FAST 2018, Vajha et al. explained the three constructions using a pairwise coupling transform. The Clay (coupled-layer) code was implemented and evaluated in Ceph.

Literature on High-Rate, OA MSR Codes with Optimum α

$$(n, k, d = n-1, \alpha = r^{\lceil \frac{n}{r} \rceil}),\ q \geq r\lceil \frac{n}{r} \rceil$$

- In May 2016, Ye & Barg came up with MSR codes that are based on Vandermonde RS codes.
- In July 2016, Sasidharan et.al came up with MSR codes that could be constructed from any MDS code.
- In ISIT 2017, Li et.al came up with a transformation that could convert any scalar MDS code to MSR construction.
- In FAST 2018, Vajha et al. explained the three constructions using a pairwise coupling transform. The Clay (coupled-layer) code was implemented and evaluated in Ceph.
- Sub-packetization bounds for optimal access MSR codes
  - Shown to be $\alpha \geq r^{\frac{k}{r}}$ for $d = n - 1$ by Tamo et al.
  - This bound is tightened to $\alpha \geq s^{\frac{n}{s}}$ by Balaji et al.

Literature on High-Rate, OA MSR Codes with Optimum α

$$(n, k, d = n-1, \alpha = r^{\lceil \frac{n}{r} \rceil}), \ q \geq r\lceil \frac{n}{r} \rceil$$

- In May 2016, Ye & Barg came up with MSR codes that are based on Vandermonde RS codes.
- In July 2016, Sasidharan et.al came up with MSR codes that could be constructed from any MDS code.
- In ISIT 2017, Li et.al came up with a transformation that could convert any scalar MDS code to MSR construction.
- In FAST 2018, Vajha et al. explained the three constructions using a pairwise coupling transform. The Clay (coupled-layer) code was implemented and evaluated in Ceph.
- Sub-packetization bounds for optimal access MSR codes
  - Shown to be $\alpha \geq r^{\frac{k}{r}}$ for $d = n-1$ by Tamo et al.
  - This bound is tightened to $\alpha \geq s^{\frac{n}{s}}$ by Balaji et al.
- We recently proved that the support of parity checks defining these constructions are forced by the optimal access, optimal sub-packetization.

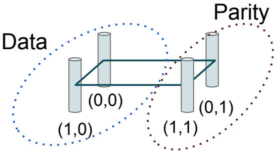
Systems Implementations of Bandwidth Efficient MDS codes

Code	MDS	Least Repair BW	Least Disk Read	Least α	Restrictions	Implemented Distributed Systems
Piggybacked RS (Sigcomm 2014)	✔	✗	✗	-	None	HDFS
Product Matrix (FAST 2015)	✔	✔	✔	✔	**Limited to Storage Overhead > 2**	Own System
Butterfly Code (FAST 2016)	✔	✔	✗	✗	Limited to the 2 parity nodes	HDFS, Ceph
HashTag Code (Trans. on Big Data 2017)	✔	✗	✗	-	Only systematic node repair	HDFS
Clay (FAST 2018)	✔	✔	✔	✔	**None!**	Ceph

- The Butterfly, HashTag codes have least disk read for systematic node repair.

Moulding an MDS Code to Yield the Clay Code

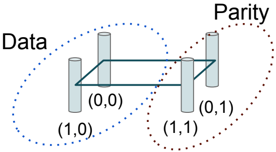
$(n = 4, k = 2)$ MDS code with optimal repair of systematic nodes, $\alpha = 2$



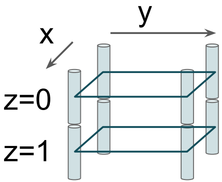
Code symbols of $[4, 2]$ MDS code.

Moulding an MDS Code to Yield the Clay Code

$(n = 4, k = 2)$ MDS code with optimal repair of systematic nodes, $\alpha = 2$



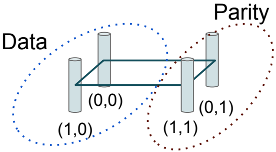
Code symbols of $[4, 2]$ MDS code.



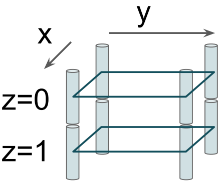
Layer two such units

Moulding an MDS Code to Yield the Clay Code

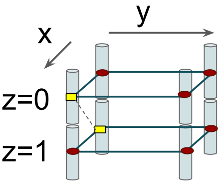
$(n = 4, k = 2)$ MDS code with optimal repair of systematic nodes, $\alpha = 2$



Code symbols of $[4, 2]$ MDS code.



Layer two such units

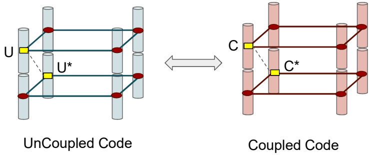


Index the layers using the red dots. symbols with yellow rectangles are paired

Uncoupled code still needs 4 symbols during recovery of single node (containing 2 symbols).

Moulding an MDS Code to Yield the Clay Code

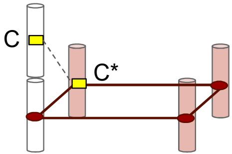
$(n = 4, k = 2)$ MDS code with optimal repair of systematic nodes, $\alpha = 2$



$$\begin{bmatrix} C \\ C^* \end{bmatrix} = \begin{bmatrix} 1 & \gamma \\ \gamma & 1 \end{bmatrix} \begin{bmatrix} U \\ U^* \end{bmatrix}$$

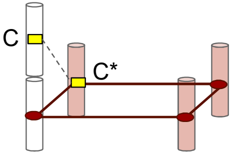
- Uncoupled code has 2 planes, where each plane corresponds to an $[4, 2]$ MDS code
- Coupled code symbols are obtained by:
  - Copying symbols with red dots
  - Pair of yellow symbols $\{C, C^*\}$ are obtained by transformation

Repair from single node loss



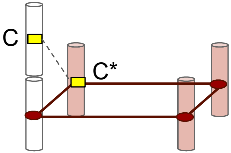
symbols that are available as part of helper information

Repair from single node loss

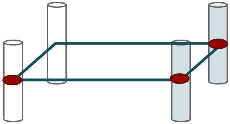


symbols that are available as part of helper information

Repair from single node loss

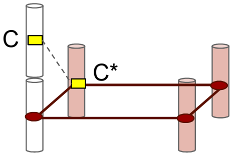


symbols that are available as part of helper information

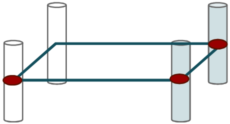


symbols that are computable in uncoupled cube

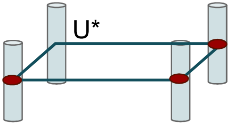
Repair from single node loss



symbols that are available as part of helper information

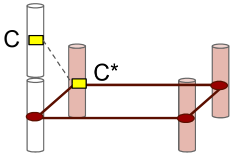


symbols that are computable in uncoupled cube

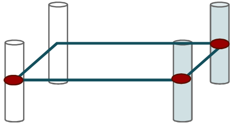


symbols recovered after using the $[4,2]$ MDS code

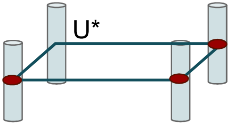
Repair from single node loss



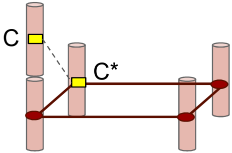
symbols that are available as part of helper information



symbols that are computable in uncoupled cube



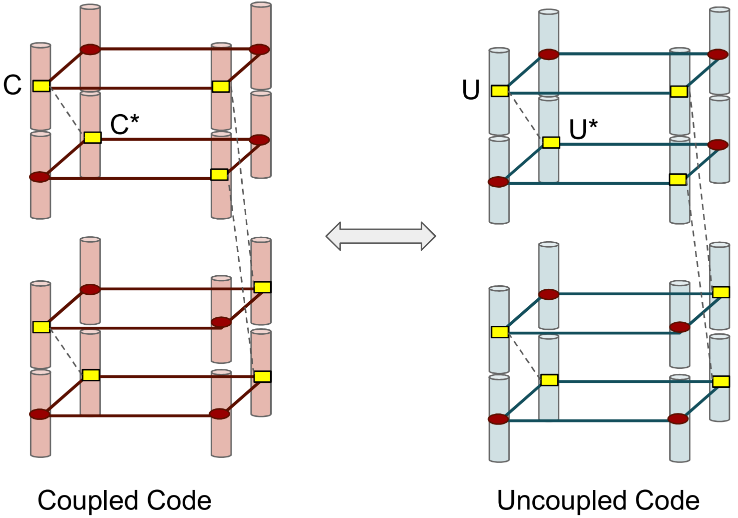
symbols recovered after using the $[4, 2]$ MDS code



C recovered from C^*, U^*

Clay Code

$(n = 4, k = 2, d = 3)$ MSR code with all node optimal repair



- The same construction extends to any (n, k, d)

Open Source: Contributions

A popular opensource distributed storage system used by CERN, Flipkart, Cisco etc

Open Source: Contributions

A popular opensource distributed storage system used by CERN, Flipkart, Cisco etc

"Us (+Vinayak) pitching Clay codes to Ceph in April 2017"

We have introduced Clay code as erasure code plugin. It is part of Ceph's Nautilus release (March 2019) as experimental feature. As part of this we also introduced support for vector codes in Ceph.

Clay Code Summary

- The open-source implementation of Clay code that we provide is for any (n, k, d) parameters.

- In comparison to $(20, 16)$ RS code, for Workloads with large sized objects (64MB), the Clay code $(20, 16, 19)$:
  - resulted in repair time reduction by $3X$.
  - Improved degraded read and write performance by 27.17% and 106.68% respectively.

- Our systems work on Clay code got featured in a popular computer science blog "the morning paper"

Clay Code Summary

- The open-source implementation of Clay code that we provide is for any (n, k, d) parameters.
- In comparison to $(20, 16)$ RS code, for Workloads with large sized objects (64MB), the Clay code $(20, 16, 19)$:
  - resulted in repair time reduction by $3X$.
  - Improved degraded read and write performance by 27.17% and 106.68% respectively.
- Our systems work on Clay code got featured in a popular computer science blog "the morning paper"
- For the case when $d < n - 1$, the clay codes are not exactly MSR, though they have optimal repair bandwidth. (few compulsory helper nodes $(d - k)$ need to be contacted compulsorily during a node's recovery).

Small d MSR Construction

- MSR Construction $(d < n - 1)$ with parameters:

  $$(n = st,\ k = n - r,\ d = k + s - 1),\ (\alpha = s^t,\ \beta = s^{t-1},\ \mathbb{F}_q)$$

  for any $s \in \{2, 3, 4\},\ t \geq 2, r \geq s$.

Small d MSR Construction

- MSR Construction ($d < n-1$) with parameters:

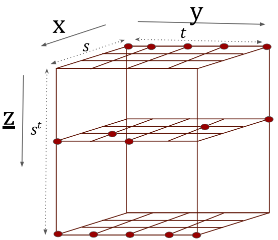
  $$(n = st,\ k = n - r,\ d = k + s - 1),\ (\alpha = s^t,\ \beta = s^{t-1},\ \mathbb{F}_q)$$

  for any $s \in \{2, 3, 4\},\ t \geq 2, r \geq s$.

- (n, k, d) MSR codes for $d \in \{k+1, k+2, k+3\}$ can be obtained by shortening $(n + \Delta, k + \Delta, d + \Delta)$ MSR code where $\Delta = \lceil \frac{n}{s} \rceil s - n$

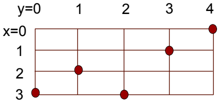
MSR Construction: 3D Representation of a Codeword

$$(n = st,\ k,\ d),\ (\alpha = s^t,\ \beta = s^{t-1},\ \mathbb{F}_q),\ s = d - k + 1$$



$s = 4,\ t = 5$

Plane dot representation

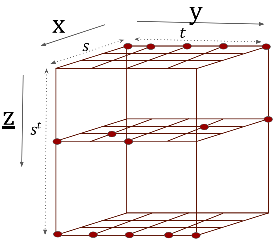


$\underline{z} = (3, 2, 3, 1, 0)$

- There are $n\alpha = s \times t \times s^t$ code symbols in $\mathbb{F}_q$.
- They can be indexed by 3-tuple $(x, y; \underline{z})$ where $x \in \mathbb{Z}_s$, $y \in \mathbb{Z}_t$, $\underline{z} \in \mathbb{Z}_s^t$.
- (x, y) tuple indicates node, $\underline{z}$ is used to index the α symbols within a node.

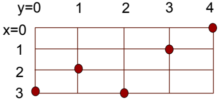
MSR Construction: 3D Representation of a Codeword

$$(n = st,\ k,\ d),\ (\alpha = s^t,\ \beta = s^{t-1},\ \mathbb{F}_q),\ s = d - k + 1$$



$s = 4,\ t = 5$

Plane dot representation



$\underline{z} = (3, 2, 3, 1, 0)$

- There are $n\alpha = s \times t \times s^t$ code symbols in $\mathbb{F}_q$.
- They can be indexed by 3-tuple $(x, y; \underline{z})$ where $x \in \mathbb{Z}_s$, $y \in \mathbb{Z}_t$, $\underline{z} \in \mathbb{Z}_s^t$.
- (x, y) tuple indicates node, $\underline{z}$ is used to index the α symbols within a node.
- The code is described by $r\alpha$ parity check equations over $n\alpha$ symbols.
  - Each parity check equation is indexed by tuple $(\ell, \underline{z})$, $\ell \in [0, r-1]$, $\underline{z} \in \mathbb{Z}_s^t$.
  - This can be viewed as r equations per plane.

MSR Construction: Parity Checks

The r parity check equations corresponding to plane $\underline{z}$ are given by:

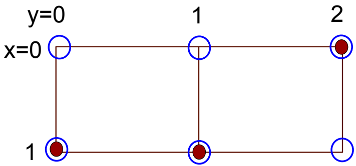
$$\underbrace{\sum_{y\in\mathbb{Z}_t}\sum_{x\in\mathbb{Z}_s}\theta^{\ell}_{x,y;z_y}C(x,y,\underline{z})}_{\text{in-plane symbols}}+\underbrace{\sum_{y\in\mathbb{Z}_t}\sum_{x\neq z_y}\gamma\theta^{\ell}_{z_y,y;x}C(z_y,y,\underline{z}(y,x))}_{\text{out-of-plane symbols}} \quad = \quad 0$$

for all $\ell \in [0, r-1]$, where $\underline{z} = (z_0, z_1, \cdots, z_{t-1})$ and

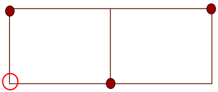
$$\underline{z}(y,x) = (z_0, \cdots, z_{y-1}, x, z_{y+1}, \cdots, z_{t-1}), \ \gamma^2 \neq 0, 1.$$

MSR Construction: Out of Plane Symbols

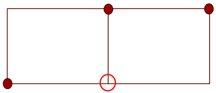
- $s = 2,\ t = 3$
- Circled symbols are involved in parity checks of plane $\underline{z} = (1,1,0)$
- Blue circles are in-plane and Red are out-of-plane



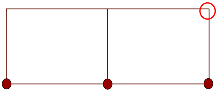
$\underline{z} = (1,1,0)$



$\underline{z}(0,0) = (0,1,0)$



$\underline{z}(1,0) = (1,0,0)$



$\underline{z}(2,1) = (1,1,1)$

MSR Construction: Theta Assignment

$$\theta_{x,y,x'} = \Theta_y(x, x'), \quad \forall x, x' \in \mathbb{Z}_s$$

Θ_y is designed to have following properties for every $y \in \mathbb{Z}_t$:

- $\theta_{x,y,x} = \theta_y$ for all $x \in \mathbb{Z}_s \implies$ to satisfy MDS property
- For $s = 2$

$$\Theta_y = \begin{bmatrix} \theta_y & \theta_{1,y} \\ \theta_{2,y} & \theta_y \end{bmatrix}$$

- For $s = 2$, need $q \geq 2n$ and for $s = 3, 4$ need $q \geq 18t + 2 = O(n)$

An Example: $s = 2, r = 3$

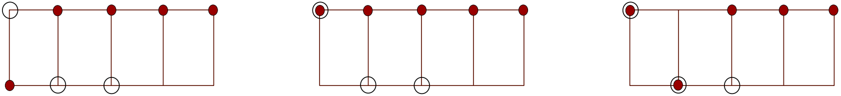
$$(n = 2t, k = n - 3, d = n - 2)$$

MDS Property:

- The code should be able to recover from any $r = 3$ erasure patterns.
- Given an r erasure pattern $\mathcal{E}$, each plane is associated with a score called Intersection Score (IS).

$$IS(\mathcal{E}, \underline{z}) = |\{(z_y, y) \in \mathcal{E} | y \in \mathbb{Z}_t\}|$$

- For, $\mathcal{E} = \{(0,0),(1,1),(1,2)\}$



IS $= 0$ IS $= 1$ IS $= 2$

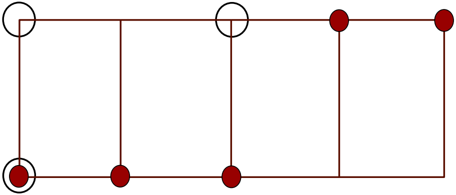
- Intersection score is number of hole-dot pairs in the plane-dot representation.

An Example: $s = 2, r = 3$

- Planes are ordered by their intersection score and erased symbols are recovered sequentially.
- Sometimes few planes with same intersection scores are to be solved together.

An Example: $s = 2, r = 3$

- Planes are ordered by their intersection score and erased symbols are recovered sequentially.
- Sometimes few planes with same intersection scores are to be solved together.
- We will look at an erasure pattern of the form:
  - Two erasures with same y value i.e., $\mathcal{E} = \{(0, y_1), (1, y_1), (x_2, y_2)\}$



Case 2

- For this case, planes can have intersection scores $1, 2$

An Example: $s = 2, r = 3$

Two erasures with same y value i.e., $\mathcal{E} = \{(0, y_1), (1, y_1), (x_2, y_2)\}$

$$\sum_{y \in \mathbb{Z}_t} \sum_{x \in \mathbb{Z}_s} \theta^{\ell}_{x,y;z_y} C(x, y, \underline{z}) + \sum_{y \in \mathbb{Z}_t} \sum_{x \neq z_y} \gamma_{x,z_y} \theta^{\ell}_{z_y,y;x} C(z_y, y, \underline{z}(y, x)) = 0$$

- $IS(\mathcal{E}, \underline{z}) = 1$, $z_{y_1} = 0$, $z_{y_2} \neq x_2$ reduces to:

$$\sum_{(x,y) \in \mathcal{E}} \theta^{\ell}_{x,y;z_y} C(x, y, \underline{z}) + \gamma_{1,0} \theta^{\ell}_{0,y_1,1} C(0, y_1, \underline{z}(y_1, 1)) = \kappa^*$$

- Look at plane $\underline{z}' = \underline{z}(y_1, 1)$, $IS(\mathcal{E}, \underline{z}) = 1$ and

$$\sum_{(x,y) \in \mathcal{E}} \theta^{\ell}_{x,y,z'_y} C(x, y, \underline{z}') + \gamma_{0,1} \theta^{\ell}_{1,y_1,0} C(1, y_1, \underline{z}) = \kappa^*$$

- 6 equations and 6 unknowns.

An Example: $s = 2, r = 3$

$$H_S = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & & \\ \theta_{0,y_1,0} & \theta_{1,y_1,0} & \theta_{x_2,y_2,a_{y_2}} & \theta_{0,y_1,1} & & \\ \theta^2_{0,y_1,0} & \theta^2_{1,y_1,0} & \theta^2_{x_2,y_2,a_{y_2}} & \theta^2_{0,y_1,1} & & \\ \hline & \gamma & & 1 & 1 & 1 \\ & \gamma\theta_{1,y_1,0} & & \theta_{0,y_1,1} & \theta_{1,y_1,1} & \theta_{x_2,y_2,a_{y_2}} \\ & \gamma\theta^2_{1,y_1,0} & & \theta^2_{0,y_1,1} & \theta^2_{1,y_1,1} & \theta^2_{x_2,y_2,a_{y_2}} \end{array}\right]$$

- Let $(f_0, f_1, f_2, g_0, g_1, g_2)^T$ be an vector in left null space of H_S. Let,

$$f(x) = \sum_{j=0}^{2} f_j x^j \text{ and } g(x) = \sum_{j=0}^{2} g_j x^j.$$

An Example: $s = 2, r = 3$

$$H_S = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & & \\ \theta_{0,y_1,0} & \theta_{1,y_1,0} & \theta_{x_2,y_2,a_{y_2}} & \theta_{0,y_1,1} & & \\ \theta^2_{0,y_1,0} & \theta^2_{1,y_1,0} & \theta^2_{x_2,y_2,a_{y_2}} & \theta^2_{0,y_1,1} & & \\ \hline & \gamma & & 1 & 1 & 1 \\ & \gamma\theta_{1,y_1,0} & & \theta_{0,y_1,1} & \theta_{1,y_1,1} & \theta_{x_2,y_2,a_{y_2}} \\ & \gamma\theta^2_{1,y_1,0} & & \theta^2_{0,y_1,1} & \theta^2_{1,y_1,1} & \theta^2_{x_2,y_2,a_{y_2}} \end{array}\right]$$

- Let $(f_0, f_1, f_2, g_0, g_1, g_2)^T$ be an vector in left null space of H_S. Let,

$$f(x) = \sum_{j=0}^{2} f_j x^j \text{ and } g(x) = \sum_{j=0}^{2} g_j x^j.$$

- $fH_s = 0$ implies that

$$\begin{aligned} f(\theta_{y_1}) = f(\theta_{x_2,y_2,a_{y_2}}) = g(\theta_{y_1}) = g(\theta_{x_2,y_2,a_{y_2}}) &= 0 \text{ where } \theta_{0,y_1,0} = \theta_{1,y_1,1} = \theta_{y_1} \\ f(\theta_{1,y_1,0}) + \gamma g(\theta_{1,y_1,0}) &= 0, \quad \gamma f(\theta_{0,y_1,1}) + g(\theta_{0,y_1,1}) = 0 \end{aligned}$$

An Example: $s = 2, r = 3$

$$H_S = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & & \\ \theta_{0,y_1,0} & \theta_{1,y_1,0} & \theta_{x_2,y_2,a_{y_2}} & \theta_{0,y_1,1} & & \\ \theta^2_{0,y_1,0} & \theta^2_{1,y_1,0} & \theta^2_{x_2,y_2,a_{y_2}} & \theta^2_{0,y_1,1} & & \\ \hline & \gamma & & 1 & 1 & 1 \\ & \gamma\theta_{1,y_1,0} & & \theta_{0,y_1,1} & \theta_{1,y_1,1} & \theta_{x_2,y_2,a_{y_2}} \\ & \gamma\theta^2_{1,y_1,0} & & \theta^2_{0,y_1,1} & \theta^2_{1,y_1,1} & \theta^2_{x_2,y_2,a_{y_2}} \end{array}\right]$$

- Let $(f_0, f_1, f_2, g_0, g_1, g_2)^T$ be an vector in left null space of H_S. Let,

$$f(x) = \sum_{j=0}^{2} f_j x^j \text{ and } g(x) = \sum_{j=0}^{2} g_j x^j.$$

- $fH_s = 0$ implies that

$$\begin{aligned} f(\theta_{y_1}) = f(\theta_{x_2,y_2,a_{y_2}}) = g(\theta_{y_1}) = g(\theta_{x_2,y_2,a_{y_2}}) &= 0 \text{ where } \theta_{0,y_1,0} = \theta_{1,y_1,1} = \theta_{y_1} \\ f(\theta_{1,y_1,0}) + \gamma g(\theta_{1,y_1,0}) &= 0, \quad \gamma f(\theta_{0,y_1,1}) + g(\theta_{0,y_1,1}) = 0 \end{aligned}$$

- Substituting $f(x) = f_2(x - \theta_{y_1})(x - \theta_{x_2,y_2,a_{y_2}})$ and $g(x) = g_2(x - \theta_{y_1})(x - \theta_{x_2,y_2,a_{y_2}})$ we get

$$\begin{bmatrix} 1 & \gamma \\ \gamma & 1 \end{bmatrix} \begin{bmatrix} f_2 \\ g_2 \end{bmatrix} = 0 \implies f_2 = g_2 = 0 \implies f = g = 0$$

MSR Codes: Summary

- Optimal access, optimal sub-packetization, explicit MSR constructions for the parameters $d = n - 1$
- Systems implementation and evaluation of Clay codes over Ceph.
- Small d constructions for $d \in \{k+1, k+2, k+3\}$.

Thanks!!