Patterned Erasure Correcting Codes for Low Storage-Overhead Blockchain Systems

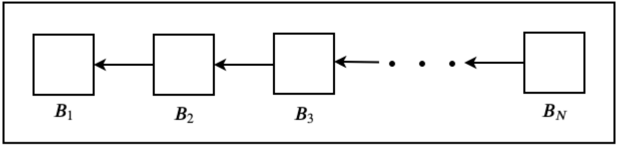
Debarnab Mitra, Lara Dolecek
Loris Lab, ECE Department, UCLA

November 6th 2019

Table of Contents

Table of Contents

Blockchain Ledger



- N blocks $B_1, B_2, \ldots, B_N$.
- Stored in the form of a hash chain $\implies$ Tamper proof

Storage Burden in Blockchains

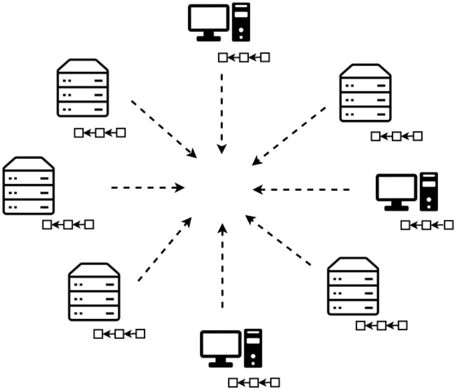


Figure: P2P Network

- P2P Network of n nodes

Storage Burden in Blockchains

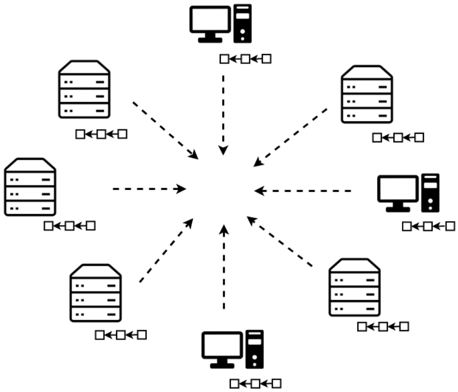


Figure: P2P Network

- P2P Network of n nodes
- Each node stores full ledger
- Decentralized

Storage Burden in Blockchains

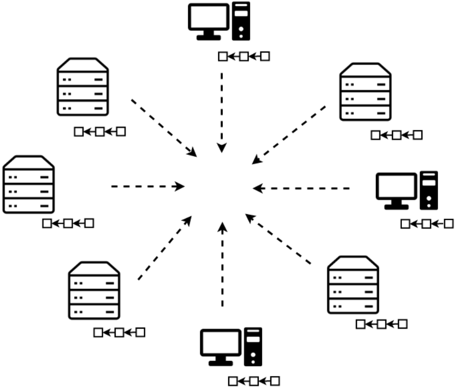


Figure: P2P Network

- P2P Network of n nodes
- Each node stores full ledger
- Decentralized
- Can correct up to $(n-1)$ node failures

Storage Burden in Blockchains

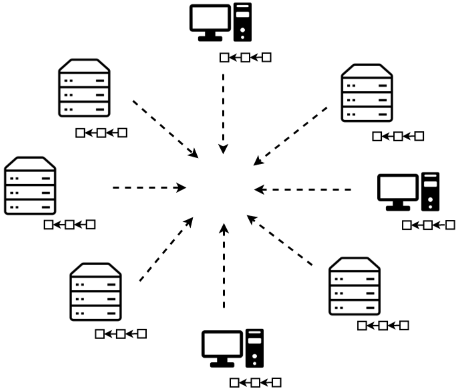


Figure: P2P Network

- P2P Network of n nodes
- Each node stores full ledger
- Decentralized
- Can correct up to $(n-1)$ node failures

Significant storage cost

Storage Burden in Blockchains

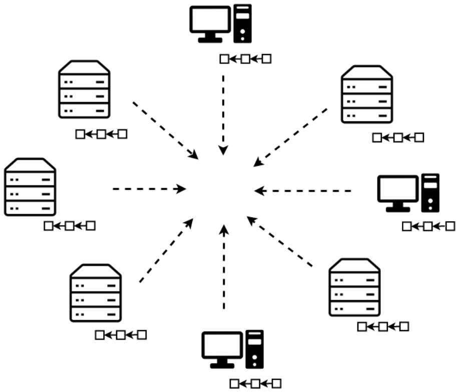


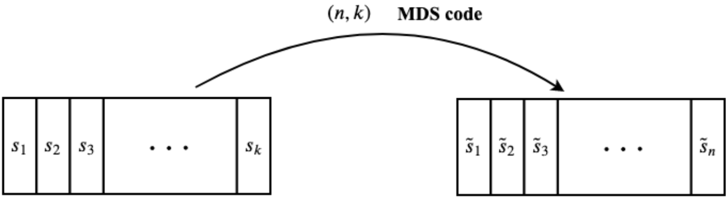
Figure: P2P Network

- P2P Network of n nodes
- Each node stores full ledger
- Decentralized
- Can correct up to $(n-1)$ node failures

Significant storage cost

Goal: Reduce storage costs without reducing blockchain availability

Prior Work: Coded Sharding[1,2]

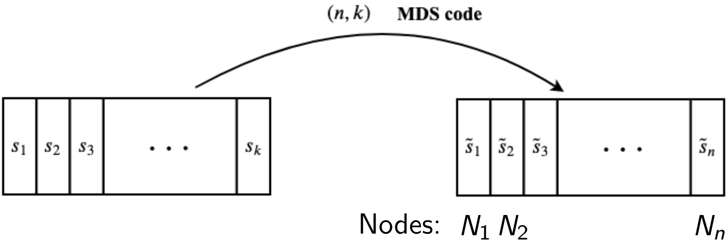


- Blockchain of size B partitioned into k shards $s_1, s_2, \ldots, s_k$
- n coded shards $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n$ generated using (n, k) MDS code

[1] M. Dai, S. Zhag, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, Mar. 2018.

[2] D. Perard, J. Lacan, Y. Bachy, and J. Detchart, "Erasure code-based low storage blockchain node," *arXiv:1805.00860*, May 2018.

Prior Work: Coded Sharding[1,2]



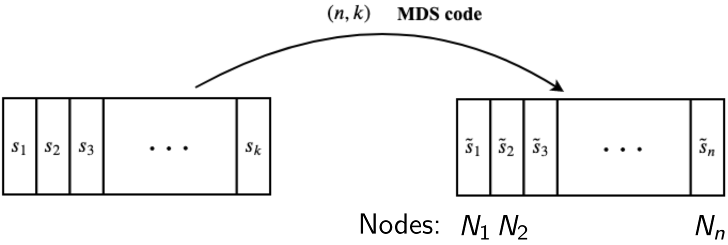
- Blockchain of size B partitioned into k shards $s_1, s_2, \ldots, s_k$
- n coded shards $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n$ generated using (n, k) MDS code
- Each node N_i stores one coded shard

---

[1] M. Dai, S. Zhag, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, Mar. 2018.

[2] D. Perard, J. Lacan, Y. Bachy, and J. Detchart, "Erasure code-based low storage blockchain node," *arXiv:1805.00860*, May 2018.

Prior Work: Coded Sharding[1,2]

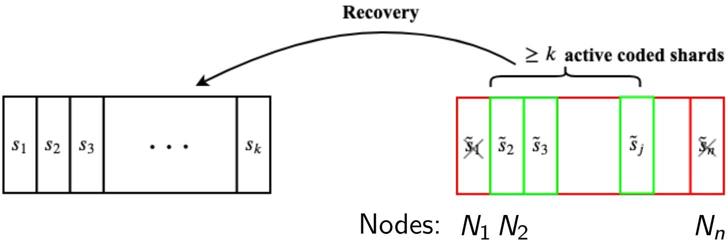


- Blockchain of size B partitioned into k shards $s_1, s_2, \ldots, s_k$
- n coded shards $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n$ generated using (n, k) MDS code
- Each node N_i stores one coded shard
- Storage at each Node: $\frac{B}{k}$,

[1] M. Dai, S. Zhag, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, Mar. 2018.

[2] D. Perard, J. Lacan, Y. Bachy, and J. Detchart, "Erasure code-based low storage blockchain node," *arXiv:1805.00860*, May 2018.

Prior Work: Coded Sharding[1,2]

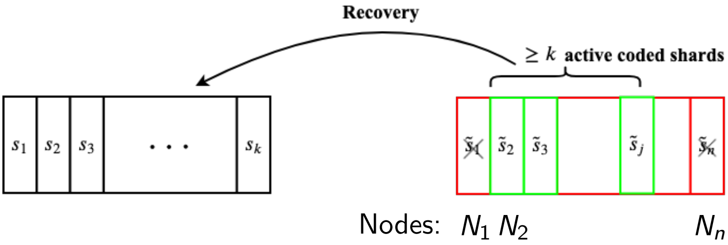


- Blockchain of size B partitioned into k shards $s_1, s_2, \ldots, s_k$
- n coded shards $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n$ generated using (n, k) MDS code
- Each node N_i stores one coded shard
- Storage at each Node: $\frac{B}{k}$,

[1] M. Dai, S. Zhag, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, Mar. 2018.

[2] D. Perard, J. Lacan, Y. Bachy, and J. Detchart, "Erasure code-based low storage blockchain node," *arXiv:1805.00860*, May 2018.

Prior Work: Coded Sharding[1,2]



- Blockchain of size B partitioned into k shards $s_1, s_2, \ldots, s_k$
- n coded shards $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n$ generated using (n, k) MDS code
- Each node N_i stores one coded shard
- Storage at each Node: $\frac{B}{k}$, corrects all $(n-k)$ node failures

[1] M. Dai, S. Zhag, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, Mar. 2018.

[2] D. Perard, J. Lacan, Y. Bachy, and J. Detchart, "Erasure code-based low storage blockchain node," *arXiv:1805.00860*, May 2018.

Table of Contents

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

$\implies$ only specific patterns of erasures possible

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities
  $\implies$ only specific patterns of erasures possible

Node 1:
Node 2:
Node 3:

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

  $\implies$ only specific patterns of erasures possible

Node 1: ✓ ✓ ✓ ×
Node 2:
Node 3:

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

$\implies$ only specific patterns of erasures possible

Node 1: $\checkmark\ \checkmark\ \checkmark\ \times\ \checkmark\ \checkmark\ \checkmark\ \times$
Node 2:
Node 3:

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network," , Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

$\implies$ only specific patterns of erasures possible

Node 1: ✓ ✓ ✓ × ✓ ✓ ✓ ×
Node 2:
Node 3:

Periodicity modelled by

- uptime, downtime (u, d)
- phase $p \in [0, u]$

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

$\implies$ only specific patterns of erasures possible

Node 1: ✓ ✓ ✓ × ✓ ✓ ✓ × ✓ ✓ ✓ ×
Node 2:
Node 3:

Periodicity modelled by

- uptime, downtime (u,d)
- phase $p \in [0, u]$

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

$\implies$ only specific patterns of erasures possible

Node 1: $\checkmark\ \checkmark\ \checkmark\ \times\ \checkmark\ \checkmark\ \checkmark\ \times\ \checkmark\ \checkmark\ \checkmark\ \times$
Node 2: $\checkmark\ \checkmark\ \times$
Node 3:

Periodicity modelled by

- uptime, downtime (u,d)
- phase $p \in [0, u]$

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities
  $\implies$ only specific patterns of erasures possible

Node 1: ✓ ✓ ✓ × ✓ ✓ ✓ × ✓ ✓ ✓ ×
Node 2: ✓ ✓ × ✓ ✓ × ✓ ✓ × ✓ ✓ ×
Node 3:

Periodicity modelled by

- uptime, downtime (u,d)
- phase $p \in [0, u]$

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

$\implies$ only specific patterns of erasures possible

Node 1:	✓ ✓ ✓ × ✓ ✓ ✓ × ✓ ✓ ✓ ×
Node 2:	✓ ✓ × ✓ ✓ × ✓ ✓ × ✓ ✓ ×
Node 3:	× × ✓ ✓ ✓ ✓ × × ✓ ✓ ✓ ✓

Periodicity modelled by

- uptime, downtime (u, d)
- phase $p \in [0, u]$

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

$\implies$ only specific patterns of erasures possible

Node 1: ✓ ✓ ✓ × ✓ ✓ ✓ × ✓ ✓ ✓ ×
Node 2: ✓ ✓ × ✓ ✓ × ✓ ✓ × ✓ ✓ ×
Node 3: × × ✓ ✓ ✓ ✓ × × ✓ ✓ ✓ ✓

Periodicity modelled by
- uptime, downtime (u, d)
- phase $p \in [0, u]$

- Patterned Erasure Set $\mathcal{P} = \{\{N_1\}, \{N_2\}, \{N_3\}, \{N_1, N_2\}, \{N_1, N_3\}\}$

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network,", Dec. 2014.

Patterned Erasures

- In practice, nodes fail periodically[3]
- Different nodes with different periodicities

  $\implies$ only specific patterns of erasures possible

Node 1: ✓ ✓ ✓ × ✓ ✓ ✓ × ✓ ✓ ✓ ×
Node 2: ✓ ✓ × ✓ ✓ × ✓ ✓ × ✓ ✓ ×
Node 3: × × ✓ ✓ ✓ ✓ × × ✓ ✓ ✓ ✓

Periodicity modelled by
- uptime, downtime (u, d)
- phase $p \in [0, u]$

- Patterned Erasure Set $\mathcal{P} = \{\{N_1\}, \{N_2\}, \{N_3\}, \{N_1, N_2\}, \{N_1, N_3\}\}$
- Can reduce storage per node by designing codes which correct only these erasure patterns

[3] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj: A peer to peer storage network," , Dec. 2014.

Goal: Storage Reduction in Patterned Erasure Model

- For a blockchain of size B with n nodes $\{N_1, N_2, \ldots N_n\}$ and erasure patterned set $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$, design a code which guarantees to corrects all erasure patterns in $\mathcal{P}$ and has minimum average storage per node

Goal: Storage Reduction in Patterned Erasure Model

- For a blockchain of size B with n nodes $\{N_1, N_2, \ldots N_n\}$ and erasure patterned set $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$, design a code which guarantees to corrects all erasure patterns in $\mathcal{P}$ and has minimum average storage per node

Lemma

Using Coded Sharding to correct all erasure patterns in $\mathcal{P}$, storage per node $\geq \frac{B}{n-t}$, where $t = \max|P_j|$.

Goal: Storage Reduction in Patterned Erasure Model

- For a blockchain of size B with n nodes $\{N_1, N_2, \ldots N_n\}$ and erasure patterned set $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$, design a code which guarantees to corrects all erasure patterns in $\mathcal{P}$ and has minimum average storage per node

Lemma

Using Coded Sharding to correct all erasure patterns in $\mathcal{P}$, storage per node $\geq \frac{B}{n-t}$, where $t = \max |P_j|$.

- Good enough to use an (n, k) MDS code which corrects all t erasure patterns

Goal: Storage Reduction in Patterned Erasure Model

- For a blockchain of size B with n nodes $\{N_1, N_2, \ldots N_n\}$ and erasure patterned set $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$, design a code which guarantees to corrects all erasure patterns in $\mathcal{P}$ and has minimum average storage per node

Lemma

Using Coded Sharding to correct all erasure patterns in $\mathcal{P}$, storage per node $\geq \frac{B}{n-t}$, where $t = \max|P_j|$.

- Good enough to use an (n, k) MDS code which corrects all t erasure patterns

Observation:

- In Coded Sharding, each node stores the same no. of shards

Goal: Storage Reduction in Patterned Erasure Model

- For a blockchain of size B with n nodes $\{N_1, N_2, \ldots N_n\}$ and erasure patterned set $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$, design a code which guarantees to corrects all erasure patterns in $\mathcal{P}$ and has minimum average storage per node

Lemma

Using Coded Sharding to correct all erasure patterns in $\mathcal{P}$, storage per node $\geq \frac{B}{n-t}$, where $t = \max|P_j|$.

- Good enough to use an (n, k) MDS code which corrects all t erasure patterns

Observation:

- In Coded Sharding, each node stores the same no. of shards

We can get a better average storage per node by relaxing this condition

Table of Contents

Optimal Shard Allocation

- Let the blockchain be partitioned into k shards $s_1, s_2, \ldots, s_k$

Optimal Shard Allocation

- Let the blockchain be partitioned into k shards $s_1, s_2, \ldots, s_k$
- Let x_i coded shards be storage at Node N_i

Optimal Shard Allocation

- Let the blockchain be partitioned into k shards $s_1, s_2, \ldots, s_k$
- Let x_i coded shards be storage at Node N_i

Average storage per node: $\frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$

Optimal Shard Allocation

- Let the blockchain be partitioned into k shards $s_1, s_2, \ldots, s_k$
- Let x_i coded shards be storage at Node N_i

Average storage per node: $\frac{B}{n} \frac{\sum_{i=1}^{n} x_i}{k}$

- Depends on k and the total number of shards stored, $\sum_{i=1}^{n} x_i$

Optimal Shard Allocation

- Let the blockchain be partitioned into k shards $s_1, s_2, \ldots, s_k$
- Let x_i coded shards be storage at Node N_i

Average storage per node: $\frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$

- Depends on k and the total number of shards stored, $\sum_{i=1}^{n} x_i$
- To minimize average storage, k and $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ should be jointly optimized

Optimal Shard Allocation

- Let the blockchain be partitioned into k shards $s_1, s_2, \dots, s_k$
- Let x_i coded shards be storage at Node N_i

Average storage per node: $\frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$

- Depends on k and the total number of shards stored, $\sum_{i=1}^{n} x_i$
- To minimize average storage, k and $\mathbf{x} = (x_1, x_2, \dots, x_n)$ should be jointly optimized

Condition for Blockchain recoverability:

- For each patterned erasure set, the number of coded shards in its complement should be at least k

Optimal Shard Allocation

- Considering k and x_i's as variables
- Code construction involves solving the following:
  (where $\bar{P}_j$ denotes the set of nodes not in P_j)

Optimal Shard Allocation

- Considering k and x_i's as variables
- Code construction involves solving the following: (where $\bar{P}_j$ denotes the set of nodes not in P_j)

Integer Optimization

$$\min_{x_1,\ldots,x_n,k} \quad \frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$$

$$s.t \sum_{i:N_i \in \bar{P}_j} x_i \geq k, j = 1, 2, \ldots, |\mathcal{P}|$$

$$x_i \in \mathrm{Z}^+, i = 1, 2, \ldots, n$$

$$k \in \mathrm{Z}^{++}$$

Optimal Shard Allocation

- Considering k and x_i's as variables
- Code construction involves solving the following: (where $\bar{P}_j$ denotes the set of nodes not in P_j)

Integer Optimization

$$\min_{x_1,\ldots,x_n,k} \quad \frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$$

$$s.t \sum_{i:N_i \in \bar{P}_j} x_i \geq k, j = 1, 2, \ldots, |\mathcal{P}|$$

$$x_i \in \mathrm{Z}^+, i = 1, 2, \ldots, n$$

$$k \in \mathrm{Z}^{++}$$

- Optimal solution $(\mathbf{x}^*, k^*)$.

Code Construction

PARE (Pattern Aware Redundancy for Erasures)- Code :

- m^{th} coded shard at N_i: $\alpha_{i,m}^{1} s_1 + \alpha_{i,m}^{2} s_2 + \ldots + \alpha_{i,m}^{k^*} s_{k^*}$, $1 \leq m \leq x_i^*$

Code Construction

PARE (Pattern Aware Redundancy for Erasures)- Code :

- m^{th} coded shard at N_i: $\alpha^1_{i,m}s_1 + \alpha^2_{i,m}s_2 + \ldots + \alpha^{k^*}_{i,m}s_{k^*}$, $1 \le m \le x^*_i$
- $\alpha^{\nu}_{i,m}$ chosen st. for each patterned set P_j, and $\{i : N_i \in \bar{P}_j\}$, the following matrix has rank k^*

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & & \\ \alpha^1_{i,1} & \alpha^2_{i,1} & \alpha^3_{i,1} & \ldots & \ldots & \alpha^{k^*}_{i,1} \\ \alpha^1_{i,2} & \alpha^2_{i,2} & \alpha^3_{i,2} & \ldots & \ldots & \alpha^{k^*}_{i,2} \\ \cdot & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & & \\ \alpha^1_{i,x^*_i} & \alpha^2_{i,x^*_i} & \alpha^3_{i,x^*_i} & \ldots & \ldots & \alpha^{k^*}_{i,x^*_i} \\ \cdot & \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \cdot & & \end{bmatrix}$$

Code Construction

PARE (Pattern Aware Redundancy for Erasures)- Code :

- m^{th} coded shard at N_i: $\alpha^1_{i,m} s_1 + \alpha^2_{i,m} s_2 + \ldots + \alpha^{k^*}_{i,m} s_{k^*}$, $1 \leq m \leq x^*_i$
- $\alpha^{\nu}_{i,m}$ chosen st. for each patterned set P_j, and $\{i : N_i \in \bar{P}_j\}$, the following matrix has rank k^*

$$\begin{bmatrix} . & . & . & . & & \\ . & . & . & . & & \\ \alpha^1_{i,1} & \alpha^2_{i,1} & \alpha^3_{i,1} & \ldots & \ldots & \alpha^{k^*}_{i,1} \\ \alpha^1_{i,2} & \alpha^2_{i,2} & \alpha^3_{i,2} & \ldots & \ldots & \alpha^{k^*}_{i,2} \\ . & . & . & . & & \\ . & . & . & . & & \\ \alpha^1_{i,x^*_i} & \alpha^2_{i,x^*_i} & \alpha^3_{i,x^*_i} & \ldots & \ldots & \alpha^{k^*}_{i,x^*_i} \\ . & . & . & . & & \\ . & . & . & . & & \end{bmatrix}$$

- Can always choose a sufficiently large field to get required $\alpha^{\nu}_{i,m}$

Code Construction

PARE (Pattern Aware Redundancy for Erasures)- Code :

- m^{th} coded shard at N_i: $\alpha^1_{i,m}s_1 + \alpha^2_{i,m}s_2 + \ldots + \alpha^{k^*}_{i,m}s_{k^*}$, $1 \leq m \leq x^*_i$
- $\alpha^{\nu}_{i,m}$ chosen st. for each patterned set P_j, and $\{i : N_i \in \bar{P}_j\}$, the following matrix has rank k^*

$$\begin{bmatrix} . & . & . & . & & \\ . & . & . & . & & \\ \alpha^1_{i,1} & \alpha^2_{i,1} & \alpha^3_{i,1} & \ldots & \ldots & \alpha^{k^*}_{i,1} \\ \alpha^1_{i,2} & \alpha^2_{i,2} & \alpha^3_{i,2} & \ldots & \ldots & \alpha^{k^*}_{i,2} \\ . & . & . & . & & \\ . & . & . & . & & \\ \alpha^1_{i,x^*_i} & \alpha^2_{i,x^*_i} & \alpha^3_{i,x^*_i} & \ldots & \ldots & \alpha^{k^*}_{i,x^*_i} \\ . & . & . & . & & \\ . & . & . & . & & \end{bmatrix}$$

- Can always choose a sufficiently large field to get required $\alpha^{\nu}_{i,m}$
- E.g. can choose $\alpha^{\nu}_{i,m}$ to form Vandermonde type matrices

Equivalence with Linear Programming

Integer Optimization

$$\min_{x_1,\ldots,x_n,k} \quad \frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$$

$$s.t \sum_{i:N_i \in \bar{P}_j} x_i \geq k, j = 1, 2, \ldots, |\mathcal{P}|$$

$$x_i \in \mathrm{Z}^{+}, i = 1, 2, \ldots, n; \; k \in \mathrm{Z}^{++}$$

$\equiv$

Equivalence with Linear Programming

Integer Optimization

$$\min_{x_1,\ldots,x_n,k} \quad \frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$$

$$s.t \sum_{i:N_i\in\bar{P}_j} x_i \geq k, j = 1, 2, \ldots, |\mathcal{P}|$$

$$x_i \in \mathrm{Z}^{+}, i = 1, 2, \ldots, n;\ k \in \mathrm{Z}^{++}$$

$\equiv$

Linear Program

$$\min_{y_1,\ldots,y_n} \sum_{i=1}^{n} y_i$$

$$s.t \sum_{i:N_i\in\bar{P}_j} y_i \geq 1, j = 1, 2, \ldots, |\mathcal{P}|$$

$$y_i \geq 0, i = 1, 2, \ldots, n$$

Equivalence with Linear Programming

Integer Optimization

$$\min_{x_1,\ldots,x_n,k} \quad \frac{B}{n}\frac{\sum_{i=1}^{n} x_i}{k}$$

$$s.t \sum_{i:N_i\in\bar{P}_j} x_i \geq k, j = 1,2,\ldots,|\mathcal{P}|$$

$$x_i \in \mathrm{Z}^+, i = 1,2,\ldots,n; \;\; k \in \mathrm{Z}^{++}$$

$\equiv$

Linear Program

$$\min_{y_1,\ldots,y_n} \sum_{i=1}^{n} y_i$$

$$s.t \sum_{i:N_i\in\bar{P}_j} y_i \geq 1, j = 1,2,\ldots,|\mathcal{P}|$$

$$y_i \geq 0, i = 1,2,\ldots,n$$

Equivalence:

- If $\mathbf{y}^* = (y_1^*, y_2^*, \ldots, y_n^*)$ is an optimal solution of the LP, then choose k^* st. $k^* \times \mathbf{y}^* = (k^* y_1^*, k^* y_2^*, \ldots, k^* y_n^*)$ is integral and $\mathbf{x}^* = k^* \times \mathbf{y}^*$
- $(\mathbf{x}^*, k^*)$ is optimal for Integer Optimization problem

PARE-Example

- 6 Nodes: $\{N_1, N_2, N_3, N_4, N_5, N_6\}$

$$\mathcal{P} = \left\{ \begin{array}{c} \{N_1, N_3, N_4, N_5\} \\ \{N_1, N_3, N_6\} \\ \{N_2, N_3, N_5, N_6\} \\ \{N_1, N_2, N_4\} \\ \{N_4, N_6\} \end{array} \right\}$$

PARE-Example

- 6 Nodes: $\{N_1, N_2, N_3, N_4, N_5, N_6\}$
- $\mathbf{y}^* = (\frac{1}{2}, \frac{1}{4}, 0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4})$
- $k^* = 4$ and $\mathbf{x}^* = (2, 1, 0, 2, 1, 3)$

$$\mathcal{P} = \left\{ \begin{matrix} \{N_1, N_3, N_4, N_5\} \\ \{N_1, N_3, N_6\} \\ \{N_2, N_3, N_5, N_6\} \\ \{N_1, N_2, N_4\} \\ \{N_4, N_6\} \end{matrix} \right\}$$

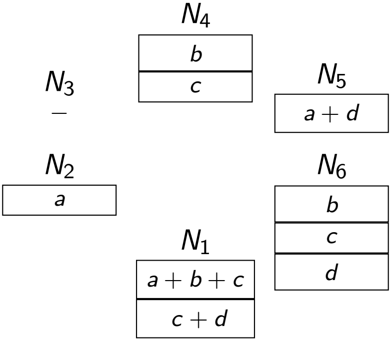
- Partition blockchain into 4 shards $\{a, b, c, d\}$

PARE-Example

- 6 Nodes: $\{N_1, N_2, N_3, N_4, N_5, N_6\}$
- $\mathbf{y}^* = (\frac{1}{2}, \frac{1}{4}, 0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4})$
- $k^* = 4$ and $\mathbf{x}^* = (2, 1, 0, 2, 1, 3)$

$$\mathcal{P} = \left\{ \begin{array}{c} \{N_1, N_3, N_4, N_5\} \\ \{N_1, N_3, N_6\} \\ \{N_2, N_3, N_5, N_6\} \\ \{N_1, N_2, N_4\} \\ \{N_4, N_6\} \end{array} \right\}$$

- Partition blockchain into 4 shards $\{a, b, c, d\}$

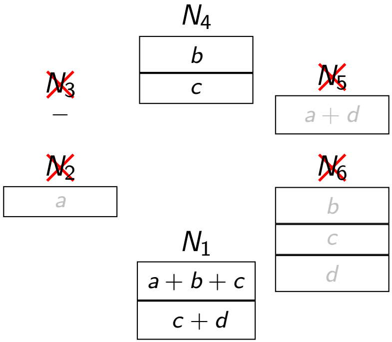


PARE-Example

- 6 Nodes: $\{N_1, N_2, N_3, N_4, N_5, N_6\}$
- $\mathbf{y}^* = (\frac{1}{2}, \frac{1}{4}, 0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4})$
- $k^* = 4$ and $\mathbf{x}^* = (2, 1, 0, 2, 1, 3)$

$$\mathcal{P} = \begin{Bmatrix} \{N_1, N_3, N_4, N_5\} \\ \{N_1, N_3, N_6\} \\ \{N_2, N_3, N_5, N_6\} \\ \{N_1, N_2, N_4\} \\ \{N_4, N_6\} \end{Bmatrix}$$

- Partition blockchain into 4 shards $\{a, b, c, d\}$

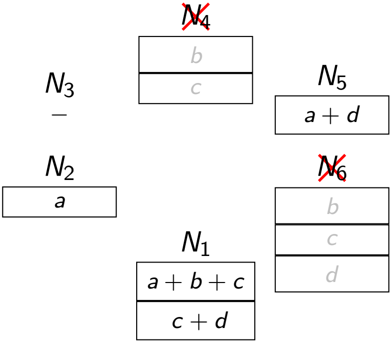


PARE-Example

- 6 Nodes: $\{N_1, N_2, N_3, N_4, N_5, N_6\}$
- $\mathbf{y}^* = (\frac{1}{2}, \frac{1}{4}, 0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4})$
- $k^* = 4$ and $\mathbf{x}^* = (2, 1, 0, 2, 1, 3)$

$$\mathcal{P} = \left\{ \begin{array}{c} \{N_1, N_3, N_4, N_5\} \\ \{N_1, N_3, N_6\} \\ \{N_2, N_3, N_5, N_6\} \\ \{N_1, N_2, N_4\} \\ \{N_4, N_6\} \end{array} \right\}$$

- Partition blockchain into 4 shards $\{a, b, c, d\}$

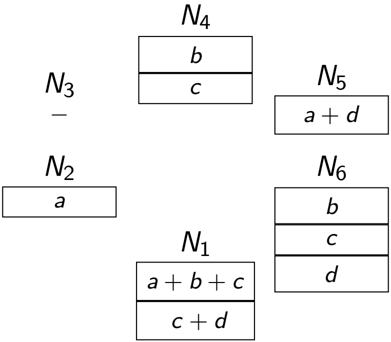


PARE-Example

- 6 Nodes: $\{N_1, N_2, N_3, N_4, N_5, N_6\}$
- $\mathbf{y}^* = (\frac{1}{2}, \frac{1}{4}, 0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4})$
- $k^* = 4$ and $\mathbf{x}^* = (2, 1, 0, 2, 1, 3)$

$$\mathcal{P} = \left\{ \begin{matrix} \{N_1, N_3, N_4, N_5\} \\ \{N_1, N_3, N_6\} \\ \{N_2, N_3, N_5, N_6\} \\ \{N_1, N_2, N_4\} \\ \{N_4, N_6\} \end{matrix} \right\}$$

- Partition blockchain into 4 shards $\{a, b, c, d\}$



- Average storage per node using PARE-Code: 0.375B
- Average storage per node using (6,2) MDS code: 0.5B

Theoretical Analysis

Lemma

Average storage per node for PARE-Code is no more than $\frac{B}{n-t}$*, where* $t = \max|P_j|$*.*

Theoretical Analysis

Lemma

Average storage per node for PARE-Code is no more than $\frac{B}{n-t}$*, where* $t = \max|P_j|$.

Proof Idea

Coded sharding with $k = n - t$ *and* $x_i = 1 \ \forall i$ *is a feasible solution and achieves an objective value of* $\frac{B}{n-t}$.

Theoretical Analysis

Lemma

Average storage per node for PARE-Code is no more than $\frac{B}{n-t}$, *where* $t = \max|P_j|$.

Proof Idea

Coded sharding with $k = n - t$ *and* $x_i = 1\ \forall i$ *is a feasible solution and achieves an objective value of* $\frac{B}{n-t}$.

Theorem

PARE-Code gives the minimum average storage per node of all codes that correct all erasure patterns in $\mathcal{P}$.

Theoretical Analysis

Lemma

Average storage per node for PARE-Code is no more than $\frac{B}{n-t}$, where $t = \max|P_j|$.

Proof Idea

Coded sharding with $k = n - t$ and $x_i = 1\ \forall i$ is a feasible solution and achieves an objective value of $\frac{B}{n-t}$.

Theorem

PARE-Code gives the minimum average storage per node of all codes that correct all erasure patterns in $\mathcal{P}$.

Proof Idea

For any coding scheme, if $B_1, B_2, \ldots, B_n$ are the amounts of the blockchain stored at $N_1, N_2, \ldots, N_n$ respectively, then B_i's must satisfy $\sum_{i:N_i \in \bar{P}_j} B_i \geq B \implies \frac{B_i}{B}$ is feasible in the LP.

Table of Contents

Effect of Adding a New Node

- (n, k) code used in Coded Sharding depends on number of nodes n

Effect of Adding a New Node

- (n, k) code used in Coded Sharding depends on number of nodes n
- Consider a system with n nodes and optimal solution $\mathbf{y}^{old}$

Effect of Adding a New Node

- (n, k) code used in Coded Sharding depends on number of nodes n
- Consider a system with n nodes and optimal solution $\mathbf{y}^{old}$
- Assume uptimes $U = [u_1, u_2, \ldots, u_r]$ and downtimes set $D = [d_1, d_2, \ldots, d_r]$.

Effect of Adding a New Node

- (n, k) code used in Coded Sharding depends on number of nodes n
- Consider a system with n nodes and optimal solution $\mathbf{y}^{old}$
- Assume uptimes $U = [u_1, u_2, \ldots, u_r]$ and downtimes set $D = [d_1, d_2, \ldots, d_r]$. Each node N_i randomly picks a $1 \leq i \leq r$ and selects the (u_i, d_i) pair and a phase $p_i \in [0, u_i]$.

Effect of Adding a New Node

- (n, k) code used in Coded Sharding depends on number of nodes n
- Consider a system with n nodes and optimal solution $\mathbf{y}^{old}$
- Assume uptimes $U = [u_1, u_2, \ldots, u_r]$ and downtimes set $D = [d_1, d_2, \ldots, d_r]$. Each node N_i randomly picks a $1 \leq i \leq r$ and selects the (u_i, d_i) pair and a phase $p_i \in [0, u_i]$.

Theorem

For the (n+1) system $\mathrm{Prob}[(\mathbf{y}^{old}, 0)$ *is optimal* $] \rightarrow 1$ *as* $n \rightarrow \infty$ *using PARE-Code.*

Effect of Adding a New Node

- (n, k) code used in Coded Sharding depends on number of nodes n
- Consider a system with n nodes and optimal solution $\mathbf{y}^{old}$
- Assume uptimes $U = [u_1, u_2, \ldots, u_r]$ and downtimes set $D = [d_1, d_2, \ldots, d_r]$. Each node N_i randomly picks a $1 \leq i \leq r$ and selects the (u_i, d_i) pair and a phase $p_i \in [0, u_i]$.

Theorem

For the (n+1) system $\mathrm{Prob}[(\mathbf{y}^{old}, 0)$ *is optimal* $] \to 1$ *as* $n \to \infty$ *using PARE-Code.*

Redesigning the coding is not needed when scaling up the number of nodes

Effect of Adding a New Node

- (n, k) code used in Coded Sharding depends on number of nodes n
- Consider a system with n nodes and optimal solution $\mathbf{y}^{old}$
- Assume uptimes $U = [u_1, u_2, \ldots, u_r]$ and downtimes set $D = [d_1, d_2, \ldots, d_r]$. Each node N_i randomly picks a $1 \leq i \leq r$ and selects the (u_i, d_i) pair and a phase $p_i \in [0, u_i]$.

Theorem

For the (n+1) system $\mathrm{Prob}[(\mathbf{y}^{old}, 0)$ *is optimal* $] \to 1$ *as* $n \to \infty$ *using PARE-Code.*

Redesigning the coding is not needed when scaling up the number of nodes

Proof Idea

For sufficiently large n, the probability that the $(n+1)^{st}$ *node has the same periodicity pattern as one of earlier nodes tends to 1.*

Condition for Redesign

LP for $(n+1)$ system:

$$\min_{y_1,\ldots,y_{n+1}} \mathbf{1}^T\mathbf{y} \qquad \max -\mathbf{b}^T\boldsymbol{\lambda}$$

$$s.t \quad \mathbf{Ay} \leq \mathbf{b} \qquad s.t \quad \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{1} = \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0}.$$

Condition for Redesign

LP for $(n+1)$ system:

$$\min_{y_1,\ldots,y_{n+1}} \mathbf{1}^T \mathbf{y} \qquad \max -\mathbf{b}^T \boldsymbol{\lambda}$$
$$s.t \quad \mathbf{Ay} \leq \mathbf{b} \qquad s.t \quad \mathbf{A}^T \boldsymbol{\lambda} + \mathbf{1} = \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0}.$$

Lemma

Let $\mathrm{I} = \{i \mid [\mathbf{y}^{old}\ 0]^T a_i = b_i\}$. $(\mathbf{y}^{old}, 0)$ *is optimal iff* $\exists \boldsymbol{\lambda} \geq \mathbf{0}$ *such that* $\mathbf{A}_{\mathrm{I}}^T \boldsymbol{\lambda} + \mathbf{1} = \mathbf{0}$.

Condition for Redesign

LP for $(n+1)$ system:

$$\min_{y_1,\ldots,y_{n+1}} \mathbf{1}^T\mathbf{y} \qquad \max -\mathbf{b}^T\boldsymbol{\lambda}$$
$$s.t \quad \mathbf{A}\mathbf{y} \leq \mathbf{b} \qquad s.t \quad \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{1} = \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0}.$$

Lemma

Let $\mathrm{I} = \{i \mid [\mathbf{y}^{old}\ 0]^T a_i = b_i\}$. $(\mathbf{y}^{old}, 0)$ *is optimal iff* $\exists\boldsymbol{\lambda} \geq \mathbf{0}$ *such that* $\mathbf{A}_{\mathrm{I}}^T\boldsymbol{\lambda} + \mathbf{1} = \mathbf{0}$.

Proof Idea

Follows from the KKT conditions on feasible point $(\mathbf{y}^{old}, 0)$.

Condition for Redesign

LP for $(n+1)$ system:

$$\min_{y_1,\ldots,y_{n+1}} \mathbf{1}^T\mathbf{y} \qquad \max -\mathbf{b}^T\boldsymbol{\lambda}$$
$$s.t \quad \mathbf{A}\mathbf{y} \leq \mathbf{b} \qquad s.t \quad \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{1} = \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0}.$$

Lemma

Let $\mathrm{I} = \{i \mid [\mathbf{y}^{old}\ 0]^T a_i = b_i\}$. $(\mathbf{y}^{old}, 0)$ *is optimal iff* $\exists \boldsymbol{\lambda} \geq \mathbf{0}$ *such that* $\mathbf{A}_{\mathrm{I}}^T\boldsymbol{\lambda} + \mathbf{1} = \mathbf{0}$.

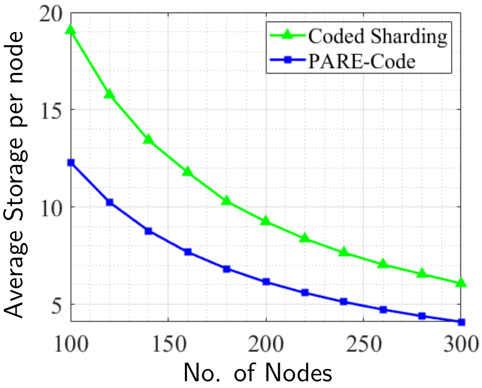
Proof Idea

Follows from the KKT conditions on feasible point $(\mathbf{y}^{old}, 0)$.

- In practice, we check this condition to decide if redesign is needed or not. With Probability 1 it is not needed.

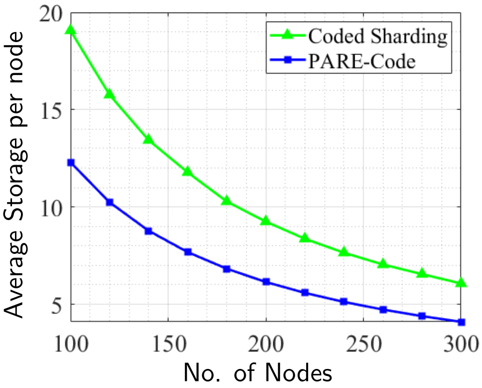
Table of Contents

Average Storage



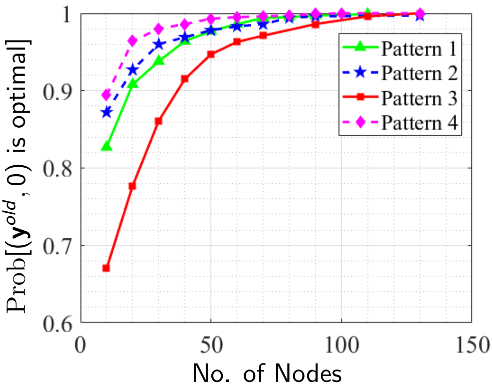
- Used $U = [5, 6, 7]$, $D = [1, 3, 5]$ and $B = 1024$

Average Storage



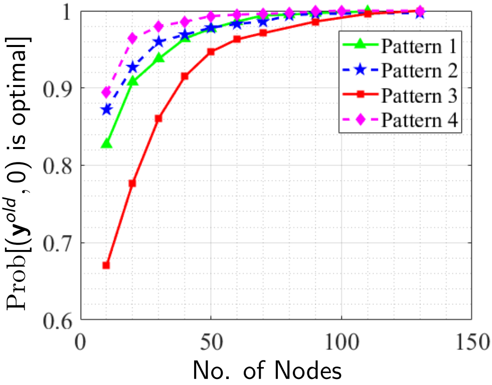
- Used $U = [5, 6, 7]$, $D = [1, 3, 5]$ and $B = 1024$
- PARE-Code has a lower average storage per node compared to coded sharding

Probability of Redesign



Pattern	U	D
1	[5,6,7]	[1,3,5]
2	[3,2,4,1,5,2]	[1,2,2,1,1,4]
3	[11,2]	[1,4]
4	[8,2]	[4,4]

Probability of Redesign



Pattern	U	D
1	[5,6,7]	[1,3,5]
2	[3,2,4,1,5,2]	[1,2,2,1,1,4]
3	[11,2]	[1,4]
4	[8,2]	[4,4]

- $\text{Prob}[(\mathbf{y}^{old}, 0)$ is optimal$] \rightarrow 1$ as number of nodes increases

Table of Contents

Conclusion and Ongoing Work

Conclusion:

- We provide a coding scheme which minimally corrects a predefined set of erasure patterns and is optimal in terms of average storage per node
- We prove that with high probability no redesign is needed using our code when there are sufficiently large number of nodes in the system

Conclusion and Ongoing Work

Conclusion:

- We provide a coding scheme which minimally corrects a predefined set of erasure patterns and is optimal in terms of average storage per node
- We prove that with high probability no redesign is needed using our code when there are sufficiently large number of nodes in the system

Ongoing Work:

- Effect of node leaving the system
- Communication cost during recovery from erasures

Thank you!