Low latency replication coded storage over memory-constrained servers

Rooji Jinan
Ajay Badita
Pradeep Sarvepalli
Parimal Parag

IEEE International Symposium on Information Theory

12-20 July 2021

Distributed Storage Systems

- How to minimize file access latency?
  - Redundancy coding

Distributed Storage Systems

- How to minimize file access latency?
  - Redundancy coding

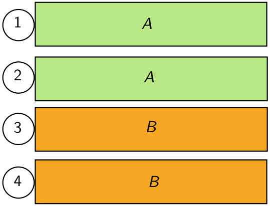


Figure: Replication code

Distributed Storage Systems

- How to minimize file access latency?
  - Redundancy coding

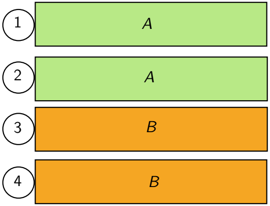


Figure: Replication code

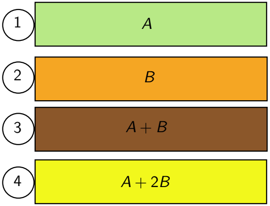
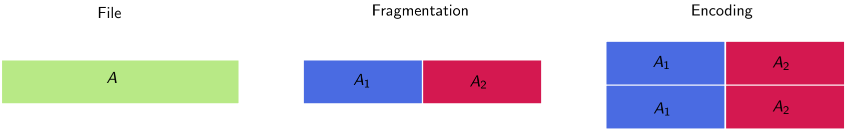


Figure: MDS code

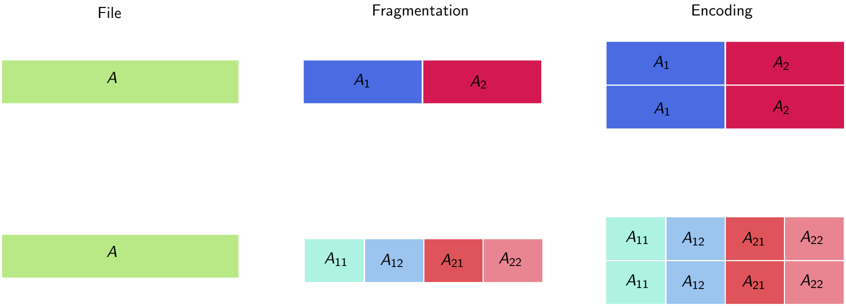
Storage model: Fragmentation & Encoding

- What if servers are memory constrained?
- File divided into V fragments & encoded into VR fragments



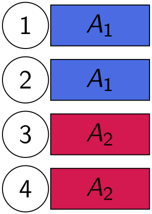
Storage model: Fragmentation & Encoding

- What if servers are memory constrained?
- File divided into V fragments & encoded into VR fragments

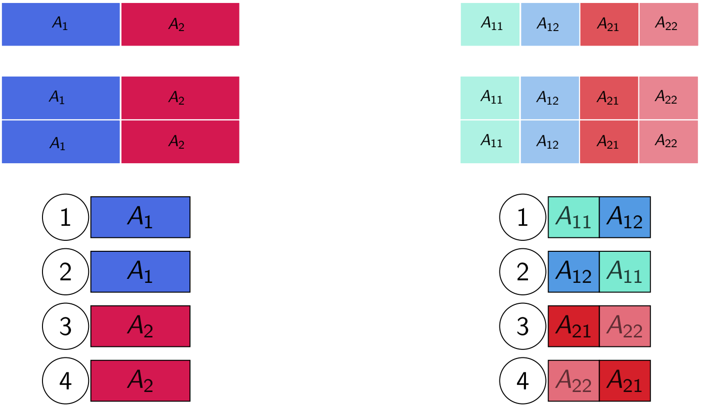


Storage model: Placement on B servers with capacity αV

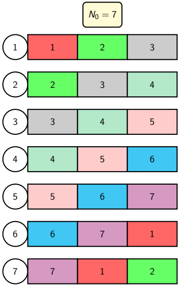
Storage model: Placement on B servers with capacity αV



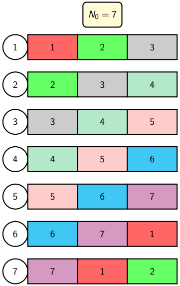
Storage model: Placement on B servers with capacity αV

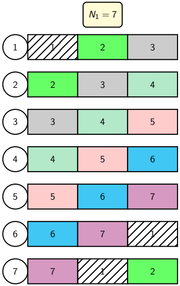


File download

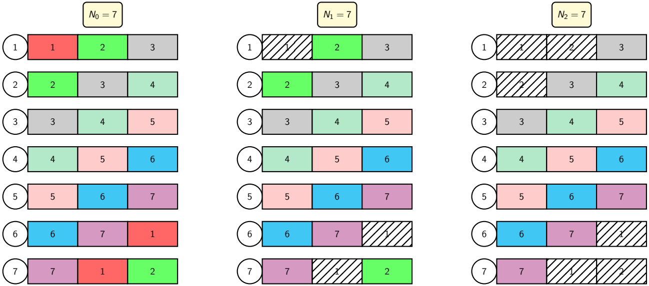


File download

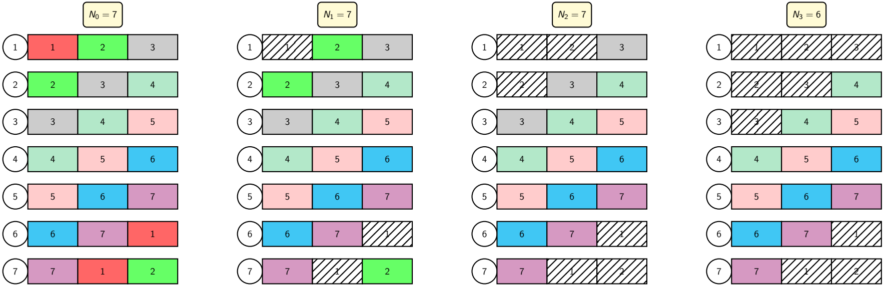




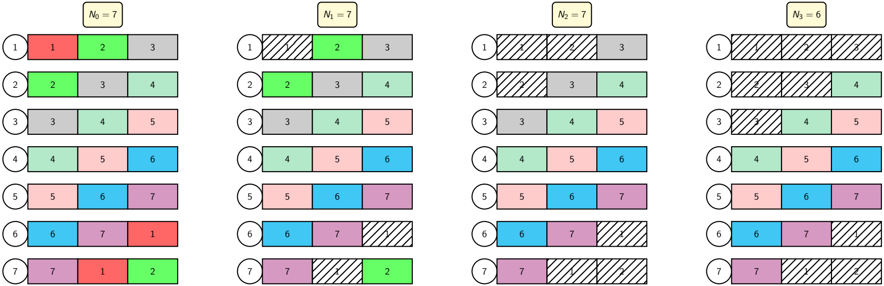
File download



File download

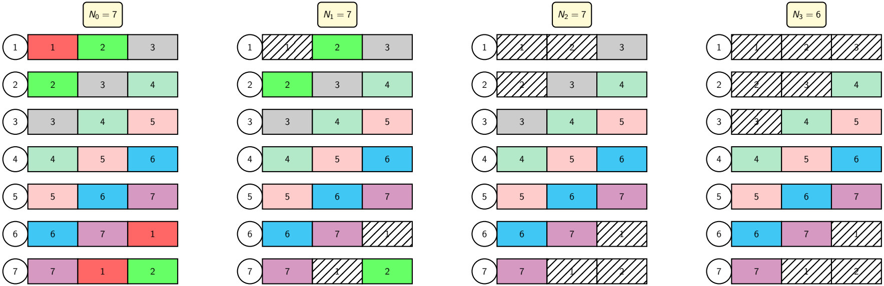


File download



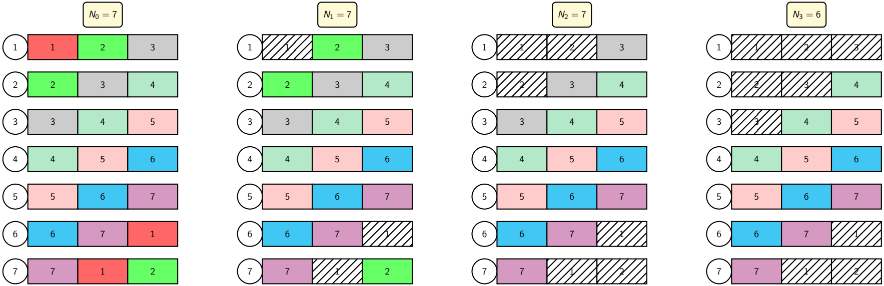
- Number of useful servers after ℓth download, N_ℓ

File download



- Number of useful servers after ℓth download, N_ℓ
- Fragment download times are i.i.d exponential with rate μ

File download



- Number of useful servers after ℓth download, N_ℓ
- Fragment download times are i.i.d exponential with rate μ
- Rate of download at ℓth stage $= N_\ell \mu$

Problem

- How to minimize the mean download time?

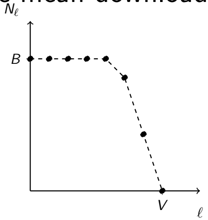


Figure: Number of useful servers number of fragments downloaded

Problem

- How to minimize the mean download time?

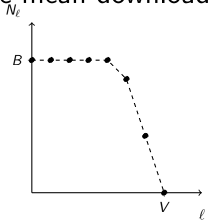


Figure: Number of useful servers number of fragments downloaded

- The normalized mean download time,

$$\frac{1}{V}\mathbb{E}\left[D_V\right] \geqslant \frac{1}{\mu \frac{1}{V}\sum_{\ell=0}^{V-1}\mathbb{E}\left[N_\ell\right]}$$

Problem

- How to minimize the mean download time?

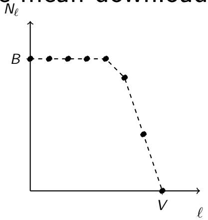


Figure: Number of useful servers number of fragments downloaded

- The normalized mean download time,

$$\frac{1}{V}\mathbb{E}\left[D_V\right] \geqslant \frac{1}{\mu\frac{1}{V}\sum_{\ell=0}^{V-1}\mathbb{E}\left[N_\ell\right]}$$

- Problem: Find a storage scheme that maximizes the mean number of useful servers averaged over all fragments.

Prior Works

MDS codes

Outperform replication codes in file access delay

- Huang et al(2012), Li et al(2016), Badita et al(2019)

Rateless codes

Offers near optimal performance

- Mallick et al(2019)

Staircase codes

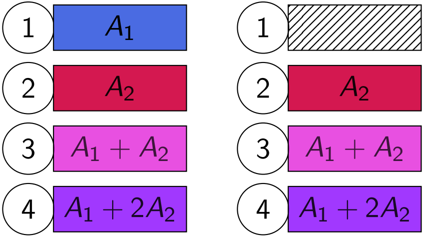
Subfragmentation improves latency performance

- Bitar et al(2020)

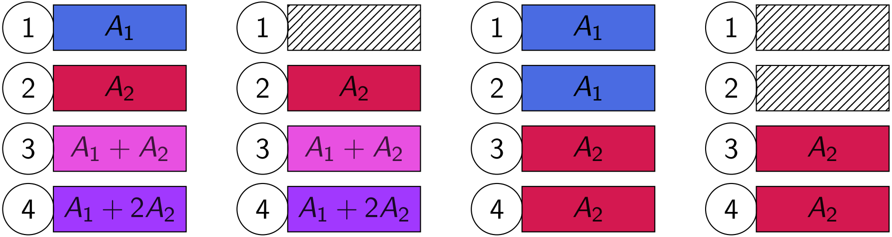
Our model

Replication codes for a file with equal sized fragmentation over multiple servers each with capacity to store more than one file fragment

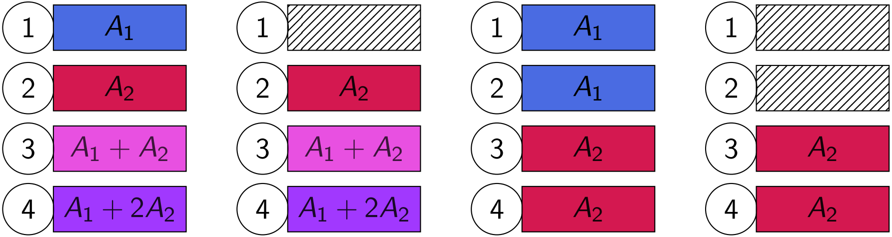
(VR, V) MDS code on α-B system



(VR, V) MDS code on α-B system



(VR, V) MDS code on α-B system

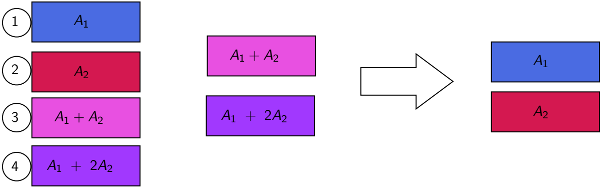


Optimality of MDS code

Reduction in useful servers is the least

Why not MDS?

- Decoding complexity



- Scaling complexity
- Need large fragment sizes

α-(V, R) replication coded storage

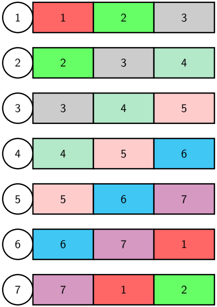


Figure: A $\frac{3}{7}$-$(7, 3)$ replication coded storage with fragment sets, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 3, 4\}$ ...

α-(V, R) replication coded storage ensemble

An α-(V, R) replication coded storage over B servers is the collection

$$\mathcal{S} \triangleq \{(S_1, S_2, \ldots, S_B) || S_b| = \alpha V \text{ for all } b, \alpha = R/B\}.$$

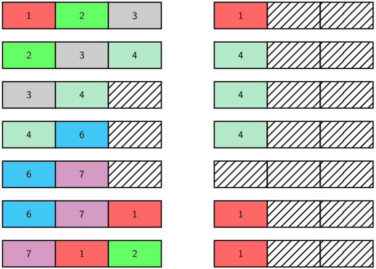
Problem statement

Problem Statement

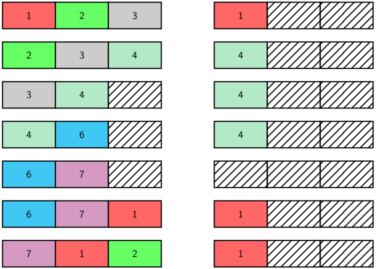
Find a storage scheme S in an α-(V, R) replication storage ensemble that maximizes the mean number of useful servers averaged over all fragments, i.e.

$$S^* = \arg\max_{S \in \mathcal{S}} \frac{1}{V} \sum_{\ell=0}^{V-1} \mathbb{E}[N_\ell].$$

A simple upper bound on N_ℓ



A simple upper bound on N_ℓ

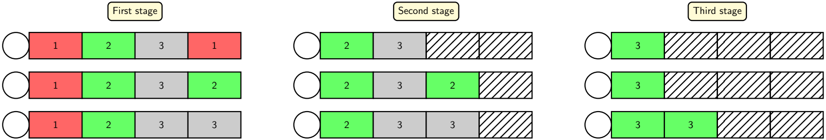


Upper Bound

For an α-(V, R) replication storage scheme, the number of useful servers N_ℓ after ℓ downloads is upper bounded in terms of $m \triangleq \lceil B/R \rceil$, as

$$N_\ell \leqslant B\mathbb{1}_{\{\ell \leqslant V-m\}} + (V-\ell)R\mathbb{1}_{\{\ell > V-m\}} \quad \& \quad \frac{1}{BV}\sum_{\ell=0}^{V-1} N_\ell \leqslant 1 - \frac{(m+1)}{2V}.$$

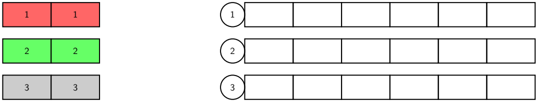
Trivial case: $\alpha \geqslant 1$



- Each server can store all the fragments
- All servers remain useful throughout
- What if $\alpha < 1$?

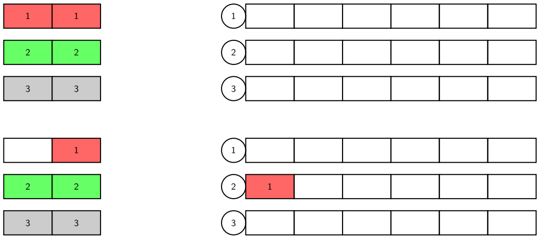
Randomized (B, V, R) replication coded storage

- Place the fragments on randomly chosen servers
- (B, V, R) replication storage scheme: Each server has the capacity to store all the VR fragments of a (VR, V) coded file



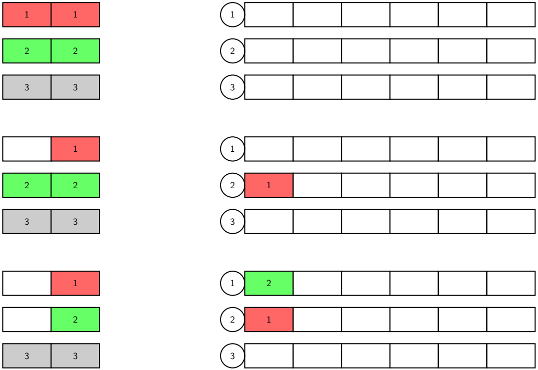
Randomized (B, V, R) replication coded storage

- Place the fragments on randomly chosen servers
- (B, V, R) replication storage scheme: Each server has the capacity to store all the VR fragments of a (VR, V) coded file



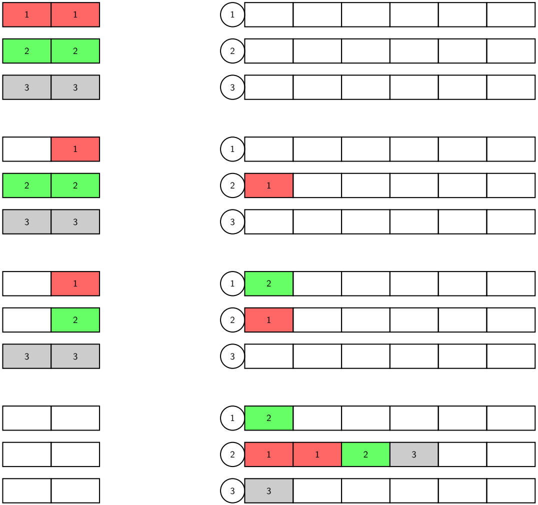
Randomized (B, V, R) replication coded storage

- Place the fragments on randomly chosen servers
- (B, V, R) replication storage scheme: Each server has the capacity to store all the VR fragments of a (VR, V) coded file



Randomized (B, V, R) replication coded storage

- Place the fragments on randomly chosen servers
- (B, V, R) replication storage scheme: Each server has the capacity to store all the VR fragments of a (VR, V) coded file

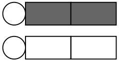


Asymptotically an α-(V, R) storage

- As V is increased with R/B fixed, the normalized number of fragments stored at any server converges to $\alpha = R/B$

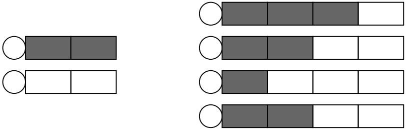
Asymptotically an α-(V, R) storage

- As V is increased with R/B fixed, the normalized number of fragments stored at any server converges to $\alpha = R/B$



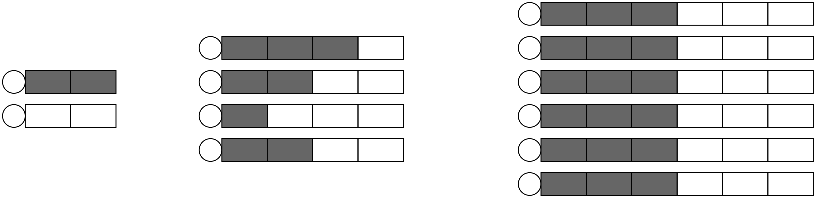
Asymptotically an α-(V, R) storage

- As V is increased with R/B fixed, the normalized number of fragments stored at any server converges to $\alpha = R/B$



Asymptotically an α-(V, R) storage

- As V is increased with R/B fixed, the normalized number of fragments stored at any server converges to $\alpha = R/B$



Theorem

The randomized (B, V, R) storage scheme is an α-(V, R) storage scheme asymptotically in V.

Performance of Random Replication Storage

Theorem

For the random (B, V, R) replication storage ensemble,

$$\frac{1}{BV}\sum_{\ell=0}^{V-1}\mathbb{E}\left[N_\ell\right] = 1 - \frac{\left(1-\frac{1}{B}\right)\left(1-(1-\frac{1}{B})^{RV}\right)}{V\left(1-(1-\frac{1}{B})^{R}\right)}.$$

Numerical Results

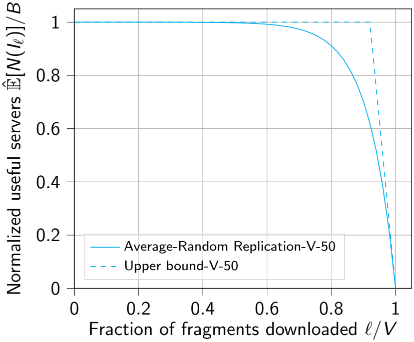


Figure: Normalized mean number of useful servers and upper bound

Numerical Results

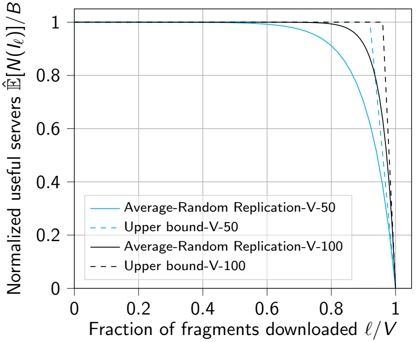


Figure: Normalized mean number of useful servers and upper bound

Numerical Results

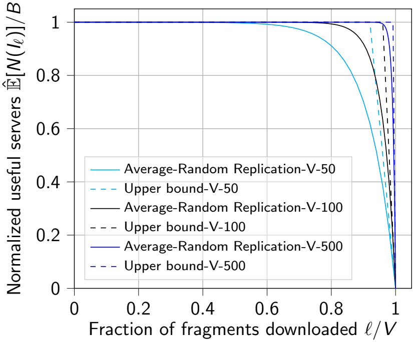


Figure: Normalized mean number of useful servers and upper bound

Numerical Results

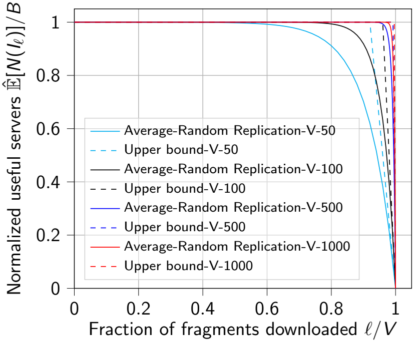


Figure: Normalized mean number of useful servers and upper bound

Numerical Results

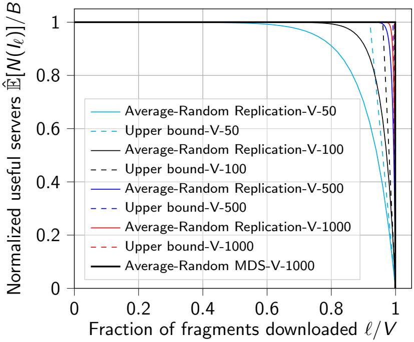


Figure: Normalized mean number of useful servers and upper bound

Number of useful servers and download times

Table: Average download times of random (B, V, R) replication storage

Random storage code			Average download time
B	V	R	
8	8	2	26936
12	12	3	14922
16	16	4	9915
20	20	5	7324

Conclusion

- We studied codes for distributed storage system with storage constraints and file subfragmentation for achieving low latency
- For exponential download times, we proposed to maximize mean number of useful servers instead of minimizing latency
- We show that MDS codes are optimal
- When there are no memory constraints at the server, replication coded file can be optimally placed
- When servers have memory constraints, we show that replication coding combined with probabilistic placement are optimal asymptotically