Low latency access of replicated fragments on memory constrained servers

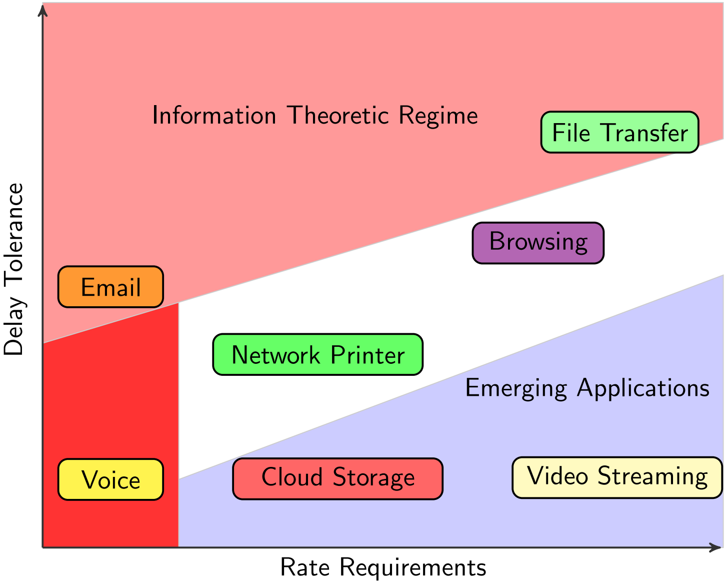
Parimal Parag
Rooji Jinan
Ajay Badita
Pradeep Sarvepalli

IMT Atlantique
Brest

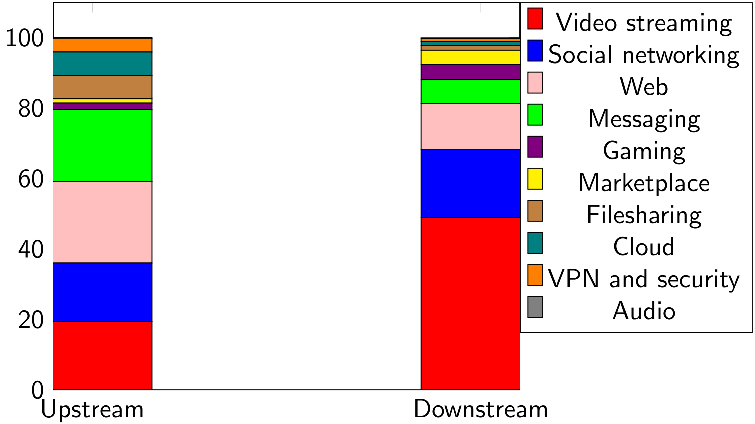
June 30, 2022



Evolving Digital Landscape

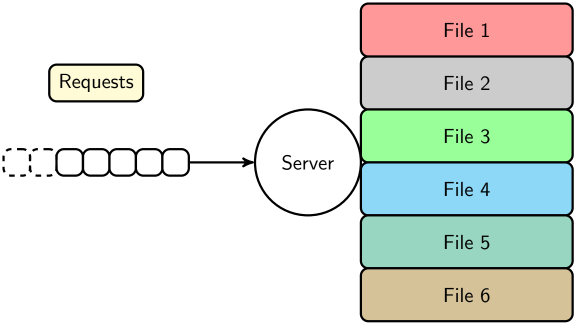


Global application traffic share 2021 [1]



[1] https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2021/Phenomena/MIPR%20Q1%202021%2020

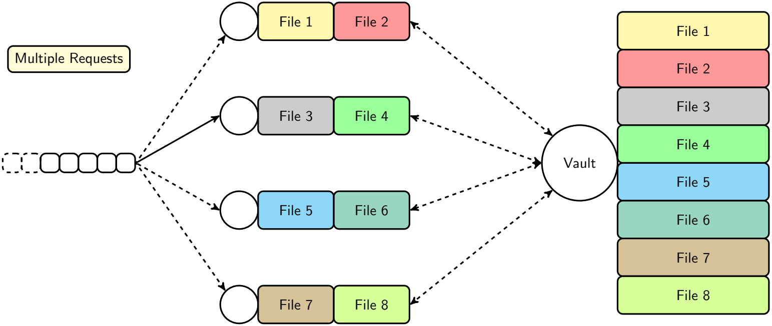
Centralized Paradigm



Potential Issues

- Not scalable with traffic load
- Susceptible to hardware failures and attacks

Distributed Paradigm



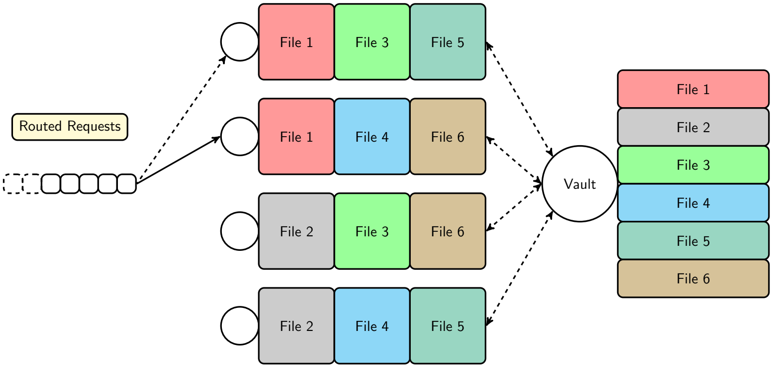
Potential Issues

- Susceptible to hardware failures and attacks

Large-scale distributed systems

- Distributed Computation
- Distributed Storage

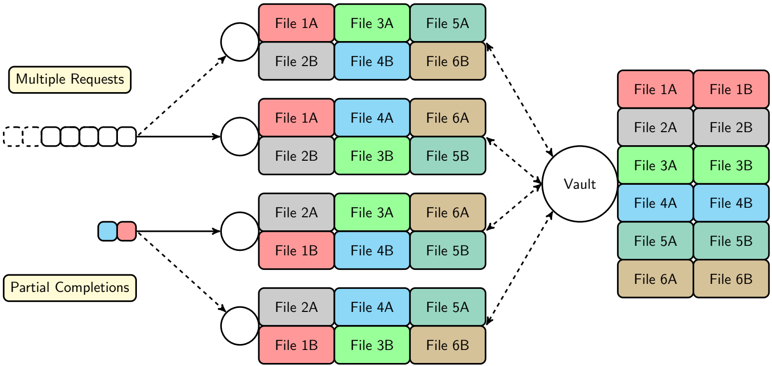
Resilience though redundancy



Latency redundancy tradeoff

- Download speedup due to parallel access
- Increased load due to redundant access

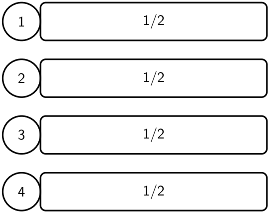
Load balancing through file fragmentation



Shared coherent access

- Availability and better content distribution
- File segments on multiple servers

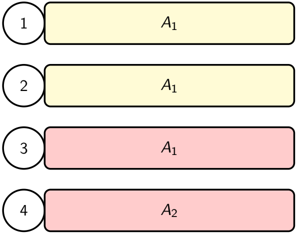
Memory constrained system

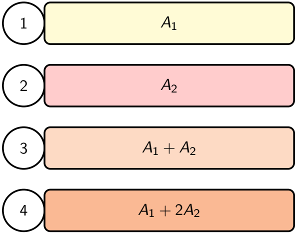


What are latency reducing storage and access schemes for replicated fragments?

- parallel access from all B servers
- α-fragment of message stored at each server

Coded Storage for single file





Single file divided into V fragments

- encoded into VR fragments
- each coded fragment stored over $B = VR$ servers
- reconstruction by set of V coded symbols

Prior Work

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

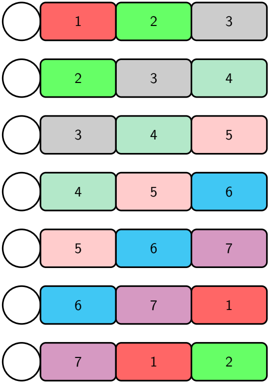
Subfragmentation improves latency performance

- Bitar et al(2020)

Our model

Replication codes for a file with equal sized fragmentation over multiple servers where each can store multiple file fragments

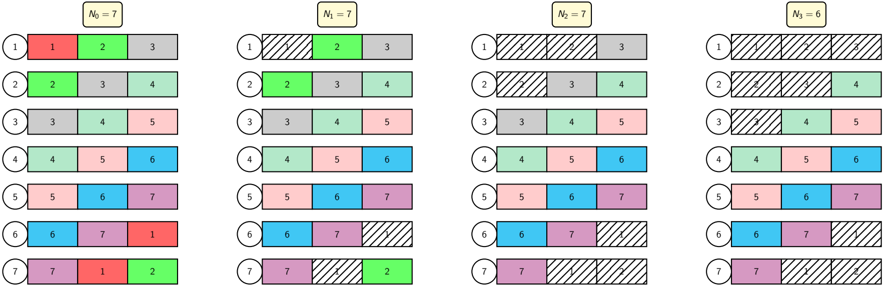
Latency optimal storage and access



A unit size divisible message $m = (m_1, \ldots, m_V)$

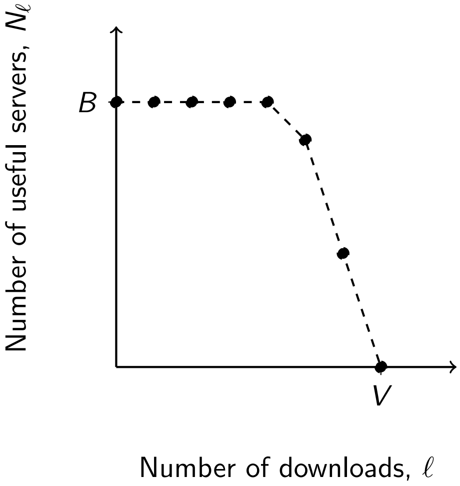
- replicated $R = \alpha B / V$ times
- **storage:** for each fragment, where to store each replica?
- **access:** for each server, sequence of access for replicas?

File download time



- Number of useful servers after ℓth download, N_ℓ
- Fragment download times are *i.i.d.* exponential with unit rate
- Rate of download at ℓth stage is N_ℓ
- The mean download time is $\mathbb{E} \sum_{\ell=0}^{V-1} \frac{1}{N_\ell}$

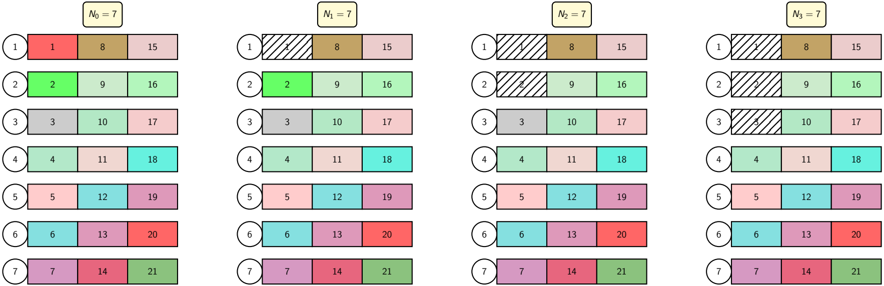
Optimality criterion



Optimality condition for storage scheme

Maximize the number of useful servers sequence

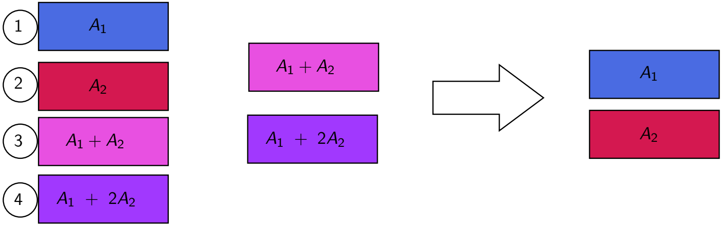
(VR, V) MDS code on α-B system



Optimality of MDS code

Reduction in useful servers is the least

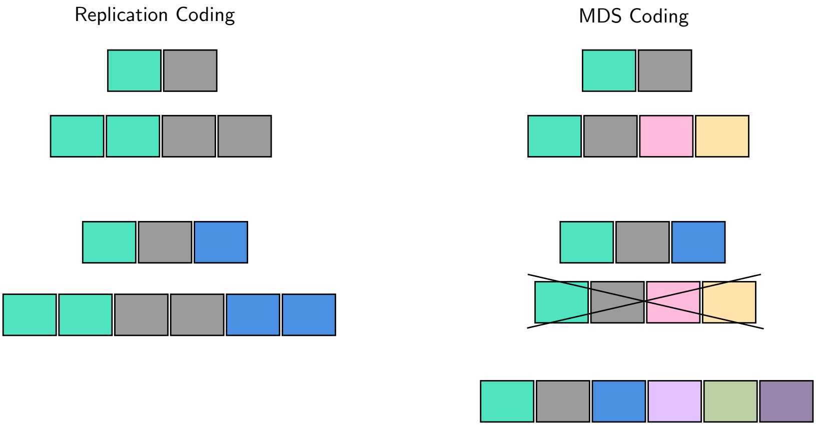
Decoding complexity



Implementation challenges

- Requires sufficiently large alphabet or large fragment sizes
- Polynomial decoding complexity that can't be parallelized

Scaling issues of MDS coding



Encoding growing data or redundancy

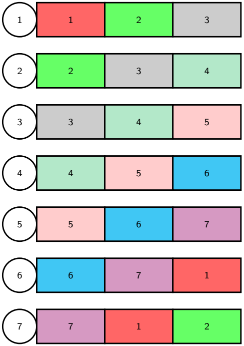
- Complete re-encoding of data blocks
- Potential data loss waiting for sufficient data blocks

Replication coded storage

α-(V, R) replication coded storage over B servers

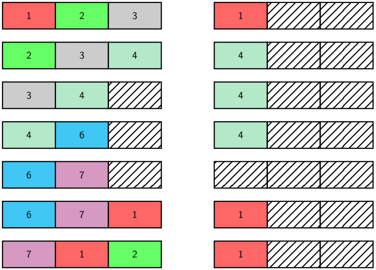
$$\mathcal{S} \triangleq \{(S_1, S_2, \ldots, S_B) : |S_b| = \alpha V \text{ for all } b, \alpha = R/B\}.$$

$\frac{3}{7} - (7, 3)$ replicated storage



- Fragment sets $S_1 = \{1, 2, 3\}, S_2 = \{2, 3, 4\}, \ldots$
- Occupancy sets $\Phi_1 = \{1, 6, 7\}, \Phi_2 = \{1, 2, 7\}, \ldots$

Upper bound on number of useful servers N_ℓ

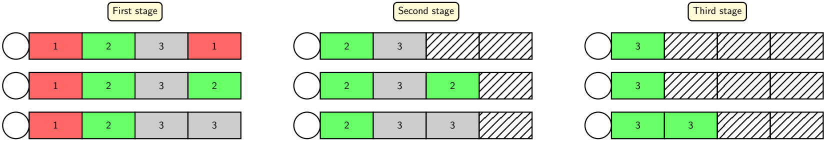


Upper bound

- For $m \triangleq \lceil B/R \rceil$, we have $N_\ell \leqslant B\mathbb{1}_{\{\ell \leqslant V-m\}} + (V-\ell)R\mathbb{1}_{\{\ell > V-m\}}$
- Normalized average of number of useful servers is upper bounded as

$$\frac{1}{BV}\sum_{\ell=0}^{V-1} N_\ell \leqslant 1 - \frac{(m+1)}{2V}.$$

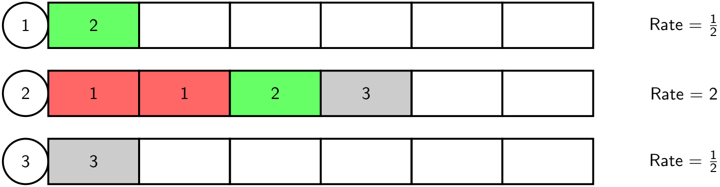
Trivial case: $\alpha \geqslant 1$



Replication as good as MDS without memory constraint

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

Randomized (B, V, R) replication coded storage

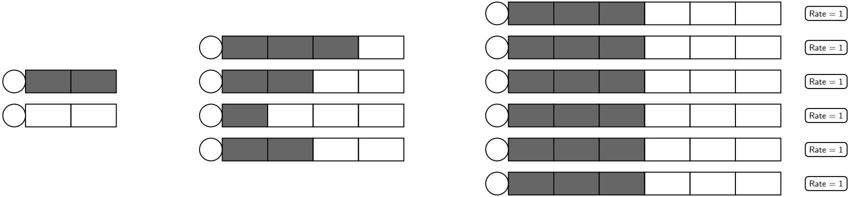


Place the fragments on randomly chosen servers

- Each server can store all coded VR fragments
- Exponential download rate $\propto$ the number of stored fragments

Asymptotically an α-(V, R) storage

- As V is increased with R/B fixed
- normalized storage at any server converges to $\alpha = R/B$
- service rate of servers converge to unity for almost all downloads



Asymptotic optimality

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

Performance of Random Replication Storage

i.i.d. random storage vector Θ where $P\{\Theta_{vr} \neq b\} = (1 - 1/B)$

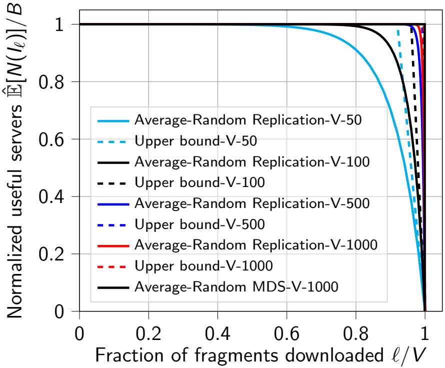
- $N_\ell = B - \sum_{b \in [B]} \prod_{v \notin I_\ell} \prod_{r \in [R]} \mathbb{1}_{\{\Theta_{vr} \neq b\}}$.
- $\frac{1}{BV}\mathbb{E}N_\ell = \frac{1}{V}\left(1 - \left(1 - \frac{1}{B}\right)^{R(V-\ell)}\right)$

Mean number of useful servers

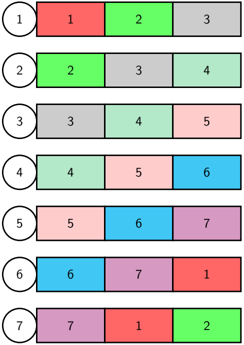
For the random (B, V, R) replication storage ensemble,

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

Numerical Results



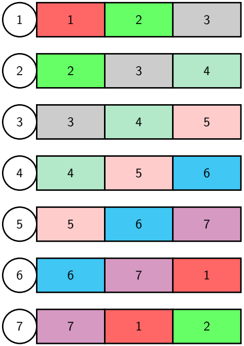
Bounding the number of useful servers



Maximum overlaps

- Between fragment sets $\tau_M \triangleq \max|S_a \cap S_b|$
- Between occupancy sets $\lambda_M \triangleq \max|\Phi_v \cap \Phi_w|$

Bounding the number of useful servers

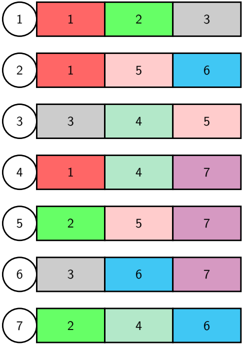


Universal bounds

- For $i \in \{0, \dots, \lfloor \frac{K}{\tau_M} \rfloor\}$ and $\ell_i \triangleq iK - i(i-1)\frac{\tau_M}{2}$

$$N_\ell \geqslant \begin{cases} B - i, & \ell_i \leqslant \ell < \ell_{i+1}, \\ (V-\ell)(R - (V - \ell - 1)\frac{\lambda_M}{2}), & \ell \geqslant V - \lfloor \frac{R}{\lambda_M} \rfloor - 1 \end{cases}$$

Optimal Storage



Tightest lower bounds

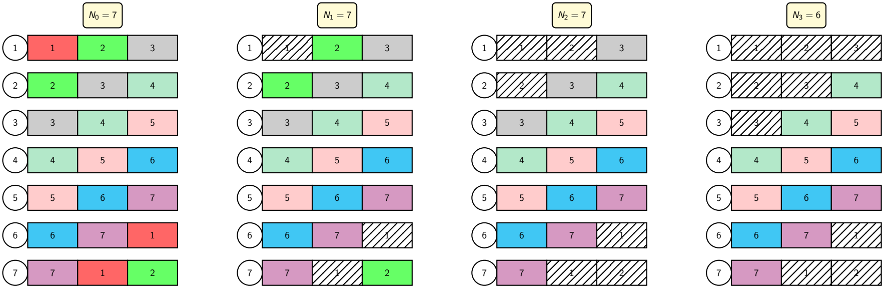
- The lower bounds are maximized for $\lambda_M = \tau_M = 1$
- Less overlaps are better

How to find the good storage schemes?

Table: Correspondence between designs and storage codes

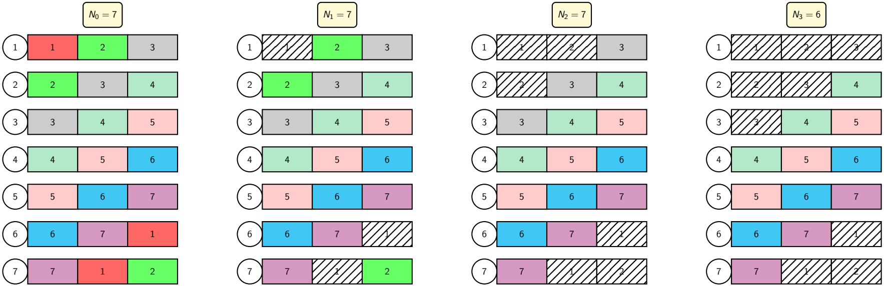
t-(V, K, λ) designs to codes	
Design parameter	**Storage parameter**
$\mathcal{P}$: Points	$[V]$: File fragments
$\mathcal{B}$: Blocks	$(S_b : b \in [B])$: Fragment sets at servers
$\lvert\mathcal{P}\rvert$: Number of points	V: Number of file fragments
$\lvert\mathcal{B}\rvert$: Number of blocks	B: Number of servers
K: Size of each block	K: Storage capacity at each server
R: Replication factor for each point	R: Replication factor for each fragment

Optimal Access



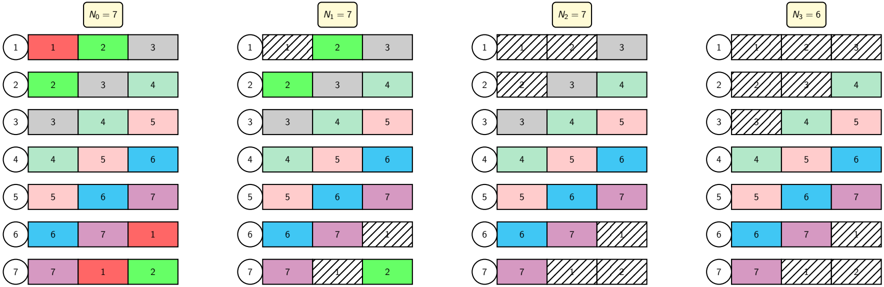
- The set of useful servers evolve as a Markov chain
- Given a storage scheme, optimal access is a Markov decision process that maximizes $\mathbb{E}\left[\sum_{\ell=0}^{V-1} N_\ell\right]$

Greedy Scheduler



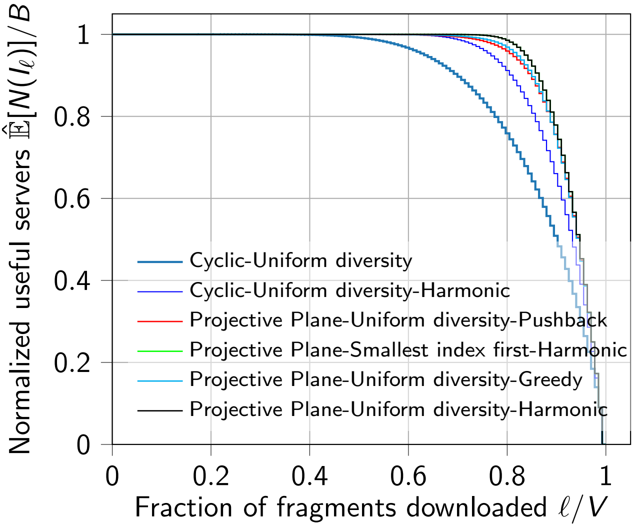
- Greedy scheduler rank $\rho^g_\ell(v) \triangleq \sum_{b\in\Phi_v} \mathbb{1}_{\{|S^\ell_b|=1\}}$ for every fragment
- $\mathbb{E}\left[N_{\ell+1} - N_\ell \mid I_\ell\right] = \sum_{v\notin I_\ell} p_{I_\ell, I_\ell\cup\{v\}}\rho^g_\ell(v)$

Ranked Scheduler



- For each useful server schedule the fragment with highest rank $\rho_\ell : I_\ell^c \to \mathbb{R}$
- Harmonic rank $\rho_\ell^h(v) \triangleq \sum_{b \in \Phi_v} \frac{1}{|S_b^\ell|}$ for every fragment

Numerical Studies



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
- Placement of coded fragments depends on overlap properties of storage codes
- Optimal access sequence is a Markov decision process

Acknowledgements

References

- R. Jinan, A. Badita, P. Sarvepalli, P. Parag. Low latency replication coded storage over memory-constrained servers. ISIT 2021.
- R. Jinan, A. Badita, P. Sarvepalli, P. Parag. Latency optimal storage and scheduling of replicated fragments for memory-constrained servers. arXiv, Sep. 2020. TIT 2022.
- A. Badita, P. Parag, and J.-F. Chamberland. Latency analysis for distributed coded storage systems. IEEE Transactions on Information Theory. 65(8):4683–4698, Aug 2019.
- Vaneet Aggarwal and Tian Lan. Modeling and optimization of latency in erasure-coded storage systems. Foundations and Trends in Communications and Information Theory. Vol. 18, Issue 3, pp 380–525, 2021.