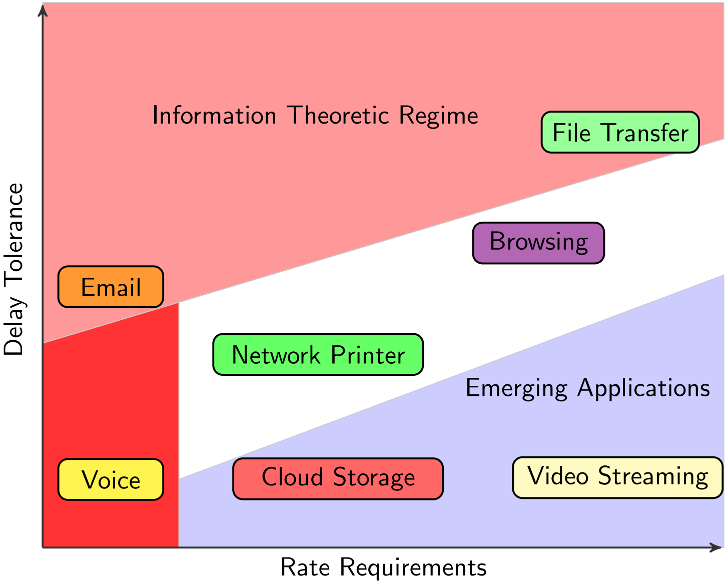
Coded parallel server systems

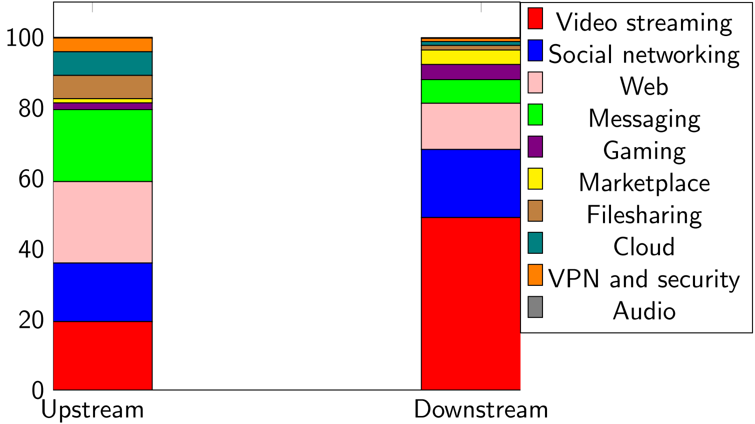
Parimal Parag

Nov 12, 2021

Evolving Digital Landscape

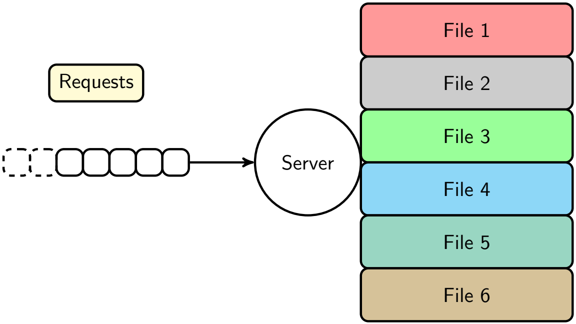


Global application traffic share 2021 [1]



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

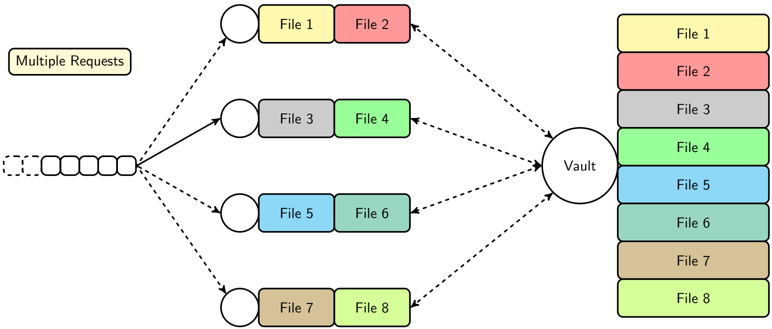
Centralized Paradigm



Potential Issues

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

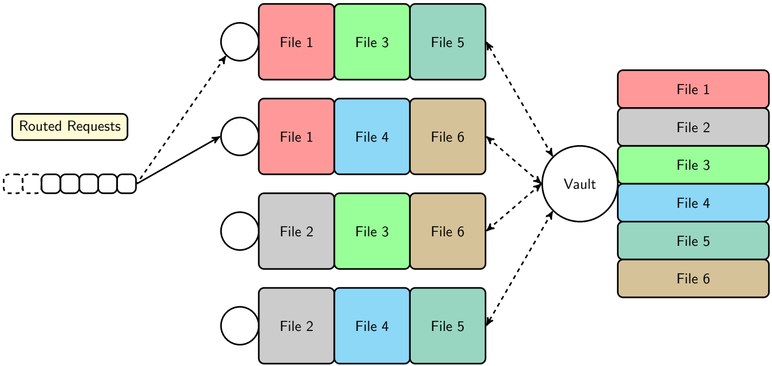
Distributed Paradigm



Potential Issues

- Susceptible to hardware failures and attacks

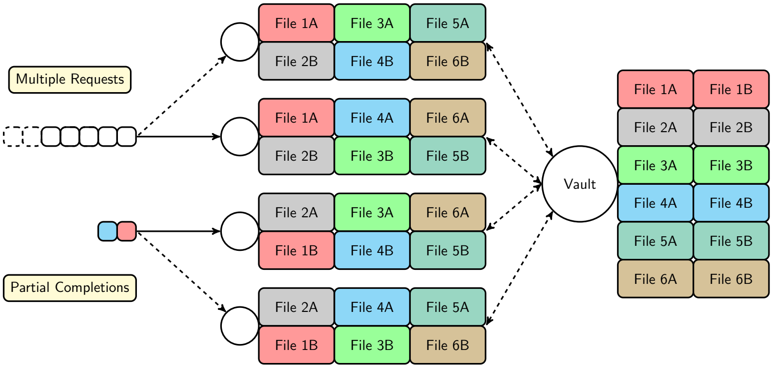
Resilience though redundancy



Latency redundancy tradeoff

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

Load balancing through file fragmentation

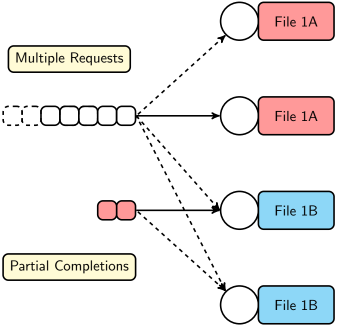


Shared coherent access

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

Independent parallel servers

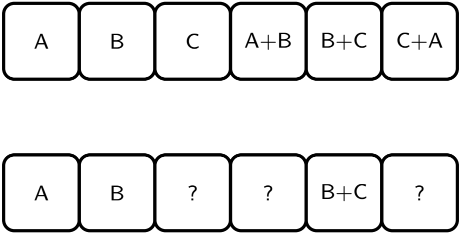
Memoryless service



Download request sent to all N parallel servers

- each server stores a single message
- query completed when K servers respond
- independent and identically distributed download times: memoryless with unit rate

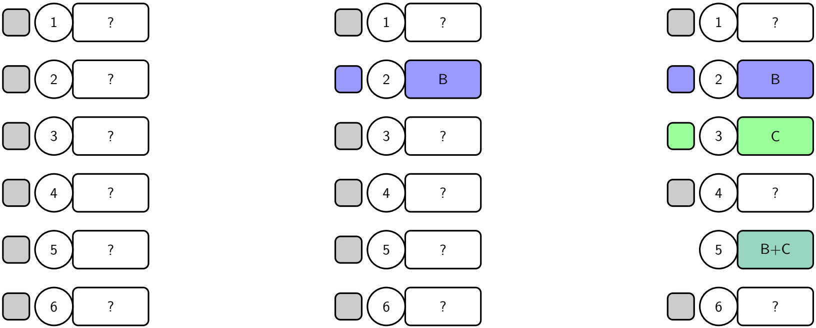
Erasure Codes



Single file divided into K fragments

- encoded into KR fragments
- each coded fragment stored over $N = KR$ servers
- reconstruction by set of K coded symbols: *information sets*

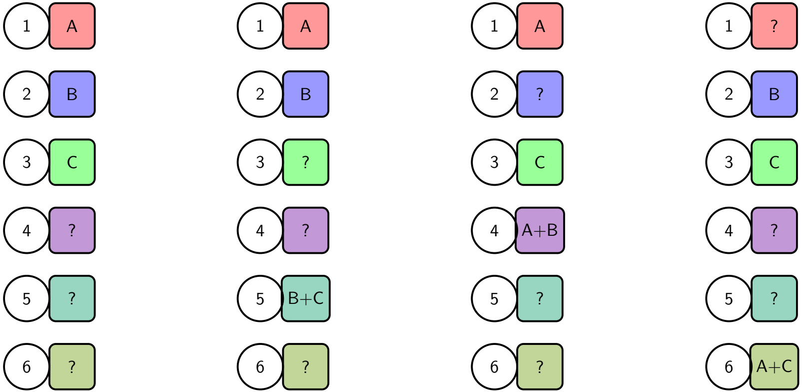
Erasure and Downloads



N coded fragments stored on N servers

- each download reveals a coded symbol
- incomplete downloads are like erased symbols
- number of erased symbols decreasing with time

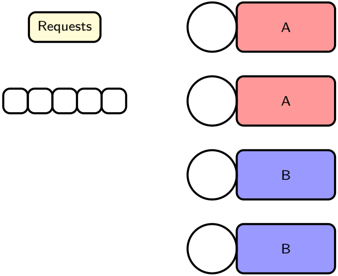
Information Sets

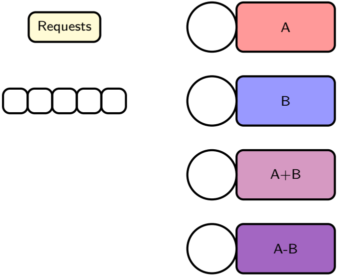


Information sets

- $\mathcal{I} = \{S \subset [n] : |S| = k, \text{ coded symbols at } S \text{ reconstruct } m\}$

Information Sets





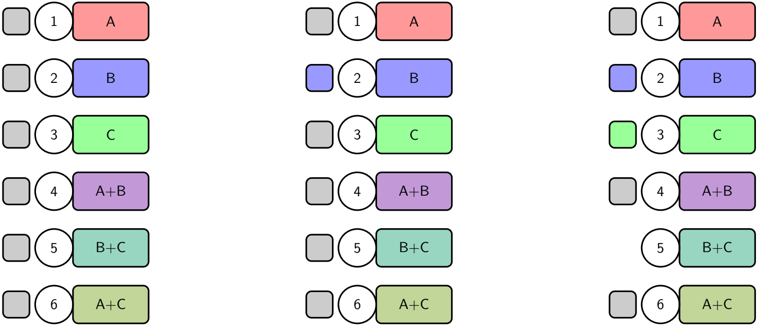
Replication (N, K)

$\mathcal{I}^{\mathrm{rep}} = \{S \subseteq [N] : |S| = K, \text{ distinct in } S\}$

MDS (N, K)

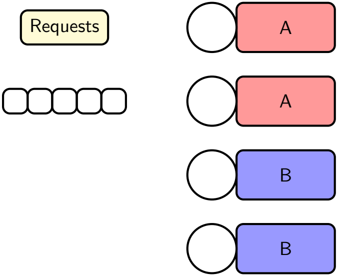
$\mathcal{I}^{\mathrm{mds}} = \{S \subseteq [N] : |S| = K\}$

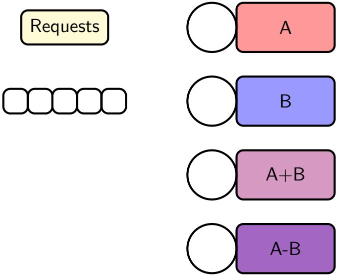
Useful Servers



- Observed servers $T \subset S$ for some info set $S \in \mathcal{I}$
- Useful servers $M(T) = \bigcup_{S \in \mathcal{I}} S \setminus T$
- **Symmetric codes:** number useful servers $N_{|T|} = |M(T)|$

Symmetric Codes





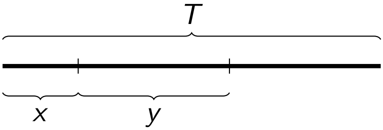
Replication (N, K)

Number of useful servers
$N_\ell = (K - \ell)N/K$

MDS (N, K)

Number of useful servers
$N_\ell = (N - \ell)$

Properties of memoryless service distributions

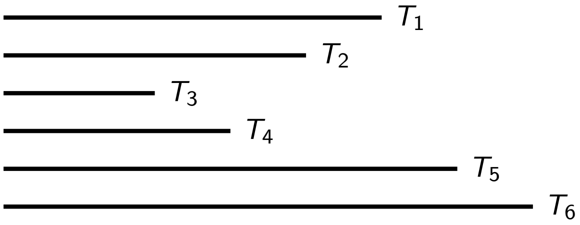


Exponential random variable T

- Tail probability $P\{T > x\} = e^{-x}$ and unit mean
- Remaining time is independent of age

$$P(\{T > x + y\} \mid \{T > x\}) = \frac{P\{T > x + y\}}{P\{T > x\}} = P\{T > y\}$$

Properties of memoryless service distributions



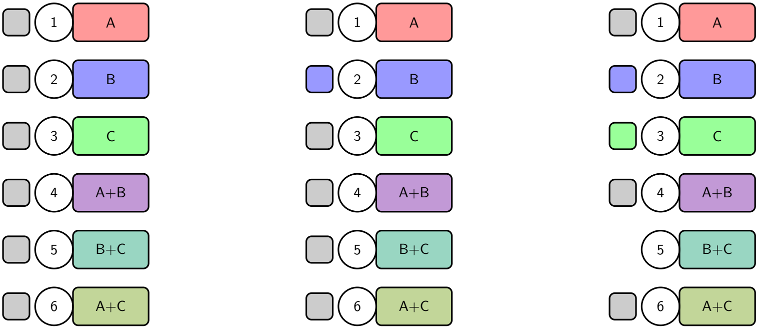
Minimum of *i.i.d.* exponential $(T_1, \ldots, T_N)$

- Minimum also exponential with rate N and hence mean $1/N$

$$P(\{\min_i T_i > x\}) = P(\cap_{i=1}^{N}\{T_i > x\}) = \prod_{i=1}^{N} P(\{T_i > x\}) = e^{-Nx}$$

- At time $T_{(1)} = \min T_i$, remaining $(N-1)$ *i.i.d.* exponential

File download time

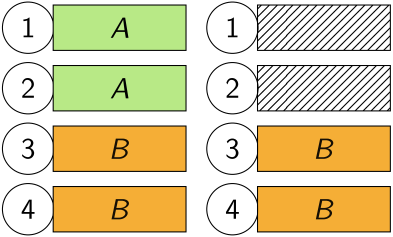


Mean file download time

- fragment downloads are *i.i.d.* and memoryless with unit rate
- parallel access from N_ℓ useful servers after ℓ downloads
- Harmonic sum of number of useful servers $\sum_{\ell=0}^{V-1} \frac{1}{N_\ell}$

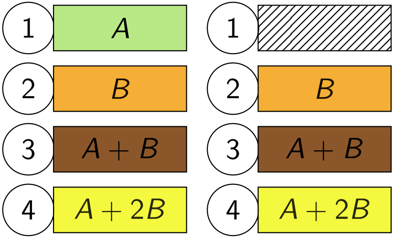
File download time

(N, K) replication code



- Mean download time $\sum_{\ell=0}^{K-1} \frac{K}{(K-\ell)N} \approx \frac{K}{N} \ln(K+1)$

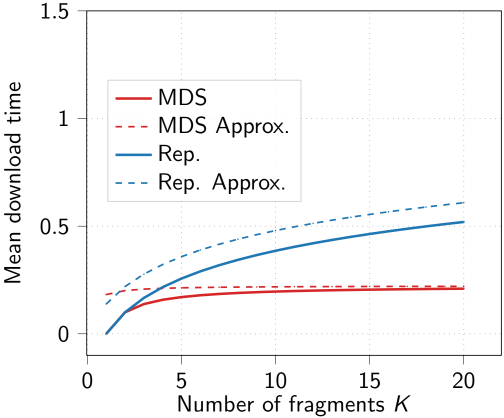
(N, K) MDS code



- Mean download time $\sum_{\ell=0}^{K-1} \frac{1}{N-\ell} \approx \frac{K}{N}$

MDS is the optimal code for minimizing the download time

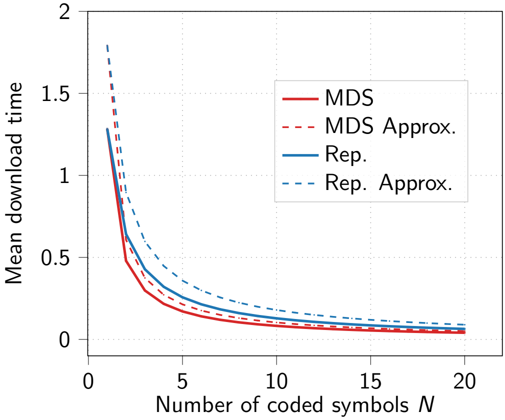
Comparison of Replication and MDS



Mean download time for code rate $\frac{K}{N} = \frac{1}{5}$

Replication performs worse as the system grows larger

Comparison of Replication and MDS



Mean download time for $K = 5$

Diminishing gains with increased redundancy and coding

Summary and Conclusion

- Reconstruction of files from the parallel download of coded fragments is similar to erasure decoding
- We computed mean download time for symmetrically coded distributed storage systems
- For exponential download times, we proposed to maximize mean number of useful servers instead of minimizing latency
- We show that MDS codes are optimal

Collaborations










Funding Agencies

References

References

- R. Jinan, A. Badita, P. Sarvepalli, P. Parag. Low latency replication coded storage over memory-constrained servers. ISIT 2021.
- S. Ramanathan, G. Gautam, V. Srinivasan, P. Parag. Latency-redundancy tradeoff in distributed read-write systems. arXiv preprint arXiv:2108.13949.
- A. Badita, R. Jinan, B. Vamanan, P. Parag. Modeling performance and energy trade-offs in online data-intensive applications. arXiv preprint arXiv:2108.08199.
- R. Jinan, A. Badita, T. Bodas, P. Parag. Load balancing policies with server-side cancellation of replicas. arXiv preprint arXiv:2010.13575.
- R. Jinan, A. Badita, P. Sarvepalli, P. Parag. Latency optimal storage and scheduling of replicated fragments for memory-constrained servers. arXiv, Sep. 2020. Under review at TIT.
- A. Badita, P. Parag, and V. Aggarwal. Single-forking of coded subtasks for straggler mitigation. IEEE/ACM Transactions on Networking.
- R. Bitar, P. Parag, and S. El Rouayheb. Minimizing latency for secure coded computing using secret sharing via staircase codes. IEEE Transactions on Communications. 68(8):4609?4619, Aug 2020.
- A. Badita, P. Parag, and V. Aggarwal. Optimal server selection for straggler mitigation. IEEE/ACM Transactions on Networking. 28(2):709?721, Apr 2020.
- 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.