On Average Latency for File Access in Distributed Coded Storage

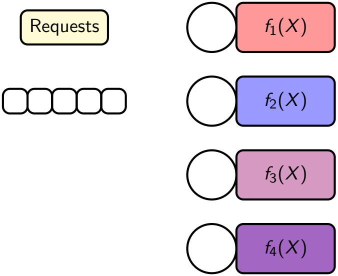
Parimal Parag
Archana Bura
Jean-François Chamberland

Electrical Communication Engineering
Indian Institute of Science

Electrical and Computer Engineering
Texas A&M University

University of Illinois, Chicago
April 26, 2017

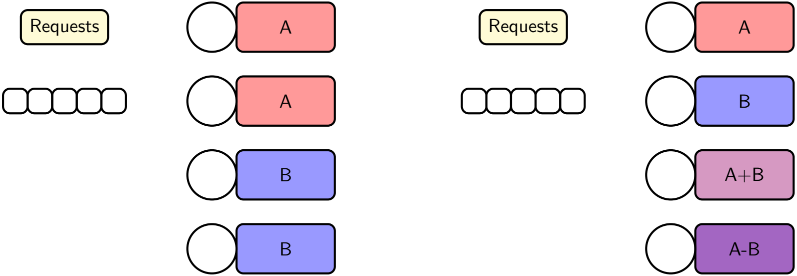
Problem Statement



Question for reducing mean access time

- How many fragments k for a single message $X = (X_1, \ldots, X_K)$?
- How to encode fragments $f_i(X)$ and store them at n distributed storage nodes?

Problem Statement



Problem

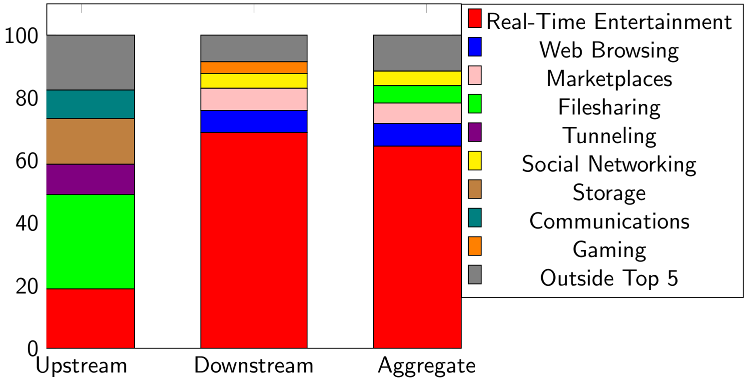
Quantify the latency gains offered by distributed coding.

Solution

Coded storage offers scaling gains over replication.

Dominant traffic on Internet

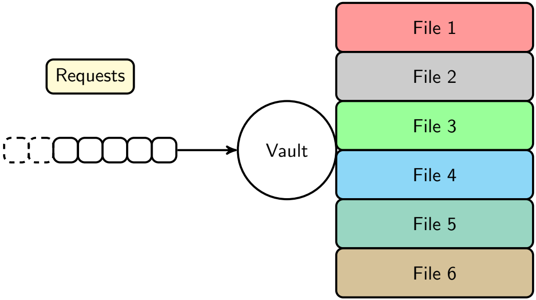
Peak Period Traffic Composition (North America)



- Real-Time Entertainment: 64.54% for downstream and 36.56 % for mobile access[1]

[1] https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/global-internet-phenomena-report-latin-america-and-north-america.pdf

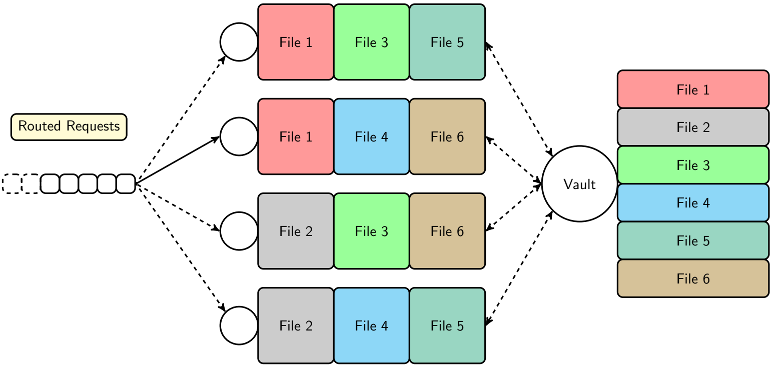
Centralized Paradigm – Media Vault



Potential Issues with Centralized Scheme

- Traffic load: Vault must handle all requests
- Service rate: Large storage entails longer access time
- Not robust to hardware failures or malicious attacks

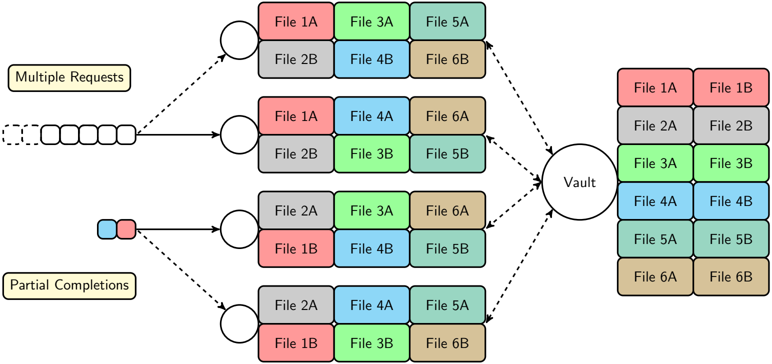
Established Solutions – Content Delivery Network



Congestion Prevention and Outage Protection

- Mirroring content with local servers
- Media file on multiple servers

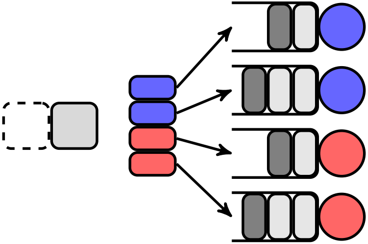
Load Balancing through File Fragmentation



Shared Coherent Access

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

Question: Duplication versus MDS Coding



Reduction of access time

- How many fragments should a single message be divided into?
- How should one encode and store at the distributed storage nodes?

System Model

File storage

- Each media file divided into k pieces
- Pieces encoded and stored on n servers

Arrival of requests

- Each request wants entire media file
- Poisson arrival of requests with rate λ

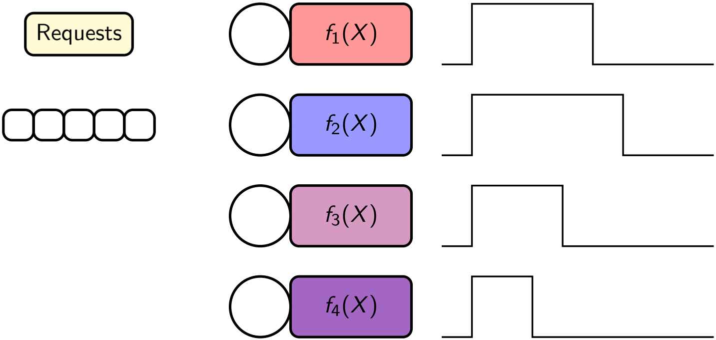
Time in the system

- Till the reception of whole file

Service at each server

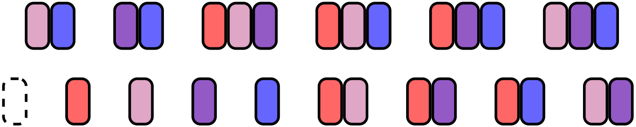
- IID exponential service time with rate k/n

Parallel Processing of Requests



- Service rate available to each request is proportional to number of servers processing the requests in parallel

State Space Structure



Keeping Track of Partially Fulfilled Requests

- Label distinct pieces with integers
- Element of state vector $Y_S(t)$ is number of users with given subets S of pieces

Continuous-Time Markov Chain

- $\mathbf{Y}(t) = \{Y_S(t) : S \subset [n]\}$ is a Markov process
- Markov process with local transitions

State Space Collapse

Theorem

For duplication and coding schemes under priority scheduling and parallel processing model, collection

$$\mathcal{S}(t) = \{S : Y_S(t) > 0, |S| < k\}$$

of information subsets is totally ordered in terms of set inclusion

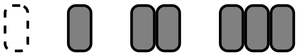
Corollary

Let $Y_i(t)$ be number of requests with i information symbols at time t, then

$$\mathbf{Y}(t) = (Y_0(t), Y_1(t), \ldots, Y_{k-1}(t))$$

is Markov process

State Transitions of Collapsed System



Arrival of Requests

- Unit increase in $Y_0(t) = Y_0(t-) + 1$ with rate λ

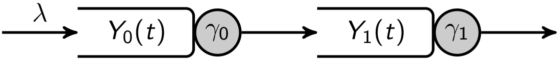
Getting Additional Symbol

- Unit increase in $Y_i(t) = Y_i(t-) + 1$
- Unit decrease in $Y_{i-1}(t) = Y_{i-1}(t-) - 1$

Getting Last Missing Symbol

- Unit decrease in $Y_{k-1}(t) = Y_{k-1}(t-) - 1$

Tandem Queue Interpretation (No Empty States)



Duplication

- When all states non-empty
- No. servers available at level i is n/k
- Normalized service rate at level i

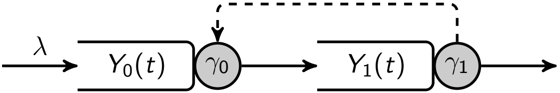
$$\gamma_i = 1 \quad i = 0, \ldots, k-1$$

MDS Coding

- When all states non-empty
- One server available at level $i \neq k-1$
- Normalized service rate at level i

$$\gamma_i = \begin{cases} \frac{k}{n} & i < k-1 \\ \frac{k}{n}(n-k+1) & i = k-1 \end{cases}$$

Tandem Queue Interpretation (General Case)



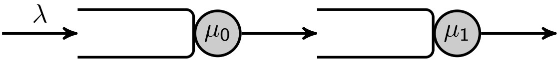
Tandem Queue with Pooled Resources

- Servers with empty buffers help upstream
- Aggregate service at level i becomes

$$\sum_{j=i}^{l_i(t)-1} \gamma_j \qquad \text{where} \qquad l_i(t) = k \wedge \{l > i : Y_l(t) > 0\}$$

- No explicit description of stationary distribution for multi-dimensional Markov process

Bounding and Separating



Theorem†

When $\lambda < \min \mu_i$, tandem queue has product form distribution

$$\pi(y) = \prod_{i=0}^{k-1} \frac{\lambda}{\mu_i} \left(1 - \frac{\lambda}{\mu_i}\right)^{y_i}$$

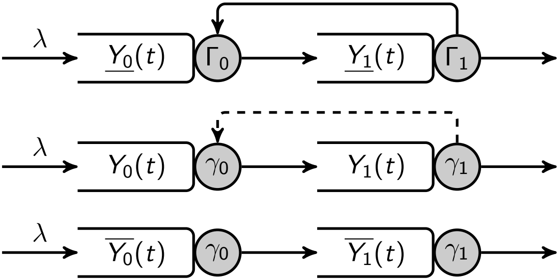
Uniform Bounds on Service Rate

Transition rates are uniformly bounded by

$$\gamma_i \leq \sum_{j=i}^{l_i(y)-1} \gamma_j \leq \sum_{j=i}^{k-1} \gamma_j \triangleq \Gamma_i$$

† F. P. Kelly, Reversibility and Stochastic Networks. New York, NY, USA: Cambridge University Press, 2011.

Bounds on Tandem Queue



Lower Bound

Higher values for service rates yield lower bound on queue distribution

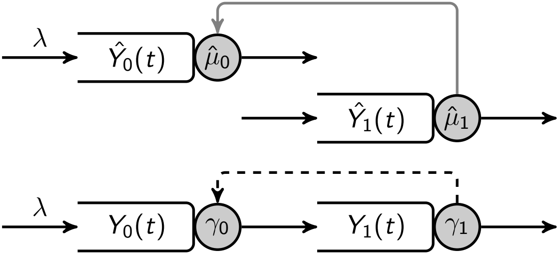
$$\underline{\pi}(y) = \prod_{i=0}^{k-1} \frac{\lambda}{\Gamma_i} \left(1 - \frac{\lambda}{\Gamma_i}\right)^{y_i}$$

Upper Bound

Lower values for service rate yield upper bound on queue distribution

$$\overline{\pi}(y) = \prod_{i=0}^{k-1} \frac{\lambda}{\gamma_i} \left(1 - \frac{\lambda}{\gamma_i}\right)^{y_i}$$

Approximating Pooled Tandem Queue



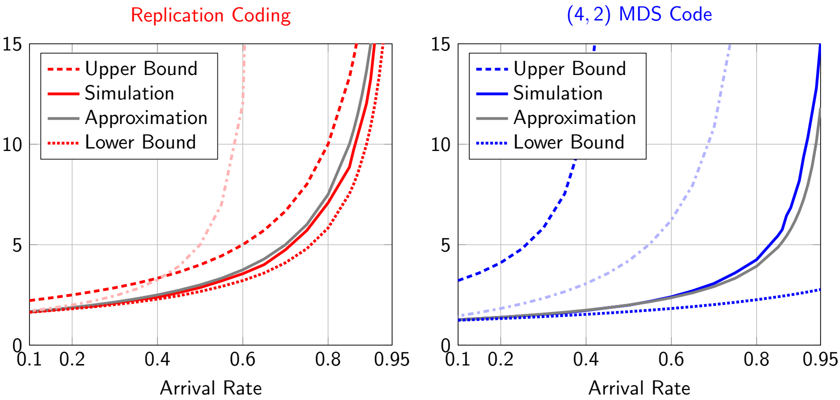
Independence Approximation with Statistical Averaging

Service rate is equal to base service rate γ_i plus cascade effect, averaged over time

$$\hat{\mu}_{k-1} = \gamma_{k-1}$$
$$\hat{\mu}_i = \gamma_i + \hat{\mu}_{i+1}\hat{\pi}_{i+1}(0)$$

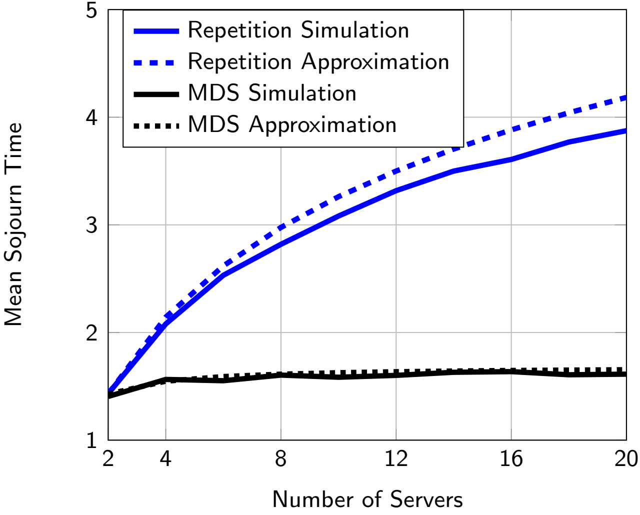
$$\hat{\pi}(y) = \prod_{i=0}^{k-1} \frac{\lambda}{\hat{\mu}_i}\left(1 - \frac{\lambda}{\hat{\mu}_i}\right)^{y_i}$$

Mean Sojourn Time



- MDS coding significantly outperforms replication
- Bounding techniques are only meaningful under light loads
- Approximation is accurate over range of loads

Comparing Replication versus MDS Coding



Arrival rate 0.3 units and coding rate $n/k = 2$

Summary and Discussion

Main Contributions

- Analytical framework for study of distributed computation and storage systems
- Upper and lower bounds to analyze replication and MDS codes
- A tight closed-form approximation to study distributed storage codes
- MDS codes are better suited for large distributed systems
- Mean access time is better for MDS codes for all code-rates