

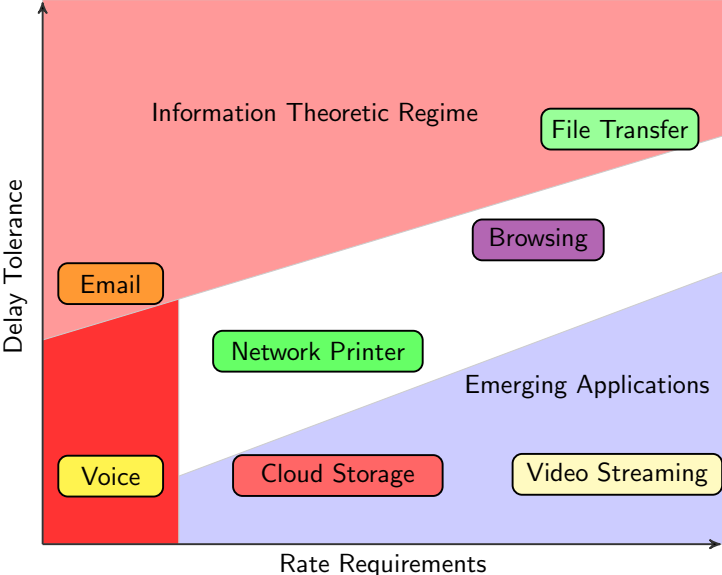
Coding in Distributed Storage and Compute Systems

Parimal Parag

Electrical Communication Engineering
Indian Institute of Science

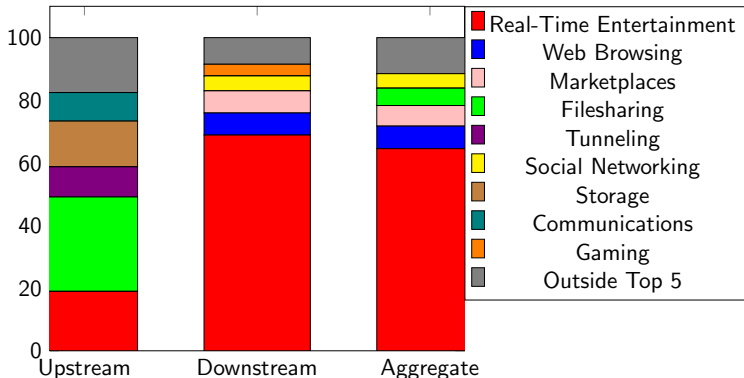
Alliance University
September 11, 2018

Evolving Digital Landscape



Dominant traffic on Internet

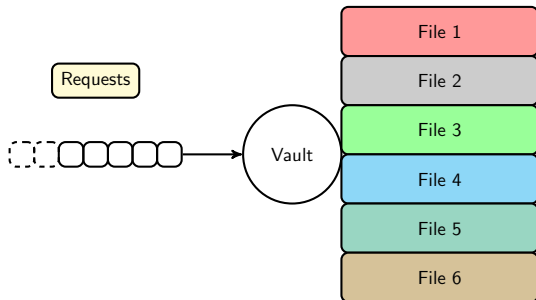
Peak Period Traffic Composition (North America)



- ▶ Real-Time Entertainment: 64.54% for downstream and 36.56% for mobile access¹

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

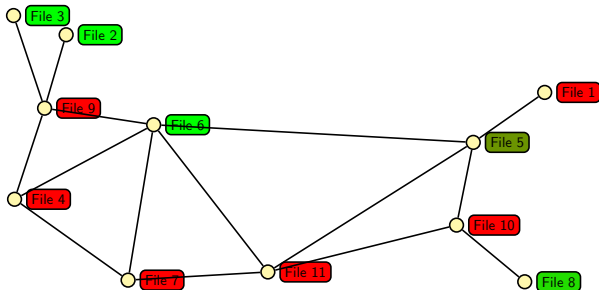
Centralized Paradigm – Media Vault



Potential Issues with Centralized Scheme

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

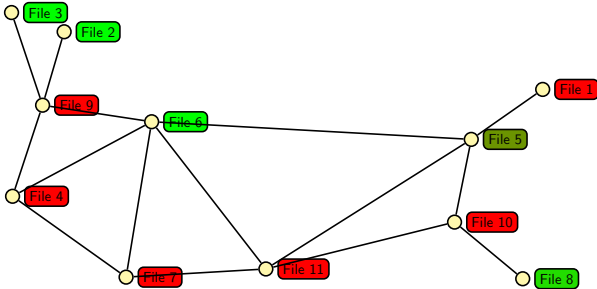
Alternative to Centralized Paradigm



Distributed Systems

- ▶ Autonomous nodes with local memory
- ▶ Interaction between the connected nodes
- ▶ Nodes with local knowledge of input and network topology
- ▶ Heterogeneous and potentially time varying system topology

Distributed Systems



Desirable Properties

- ▶ **Scalability:** Linear or sub-linear increase in number of nodes
- ▶ **Resilience:** Able to withstand local node failures
- ▶ **Efficiency:** Minimum interaction between nodes
- ▶ **Fairness:** Almost equal load at all nodes

Examples

Distributed Storage

- ▶ **Content streaming:** NetFlix, HotStar, Eros Now, YouTube, Hulu, Amazon Prime Video
- ▶ **Cloud storage:** GitHub, DropBox, iCloud, OneDrive, UbuntuOne
- ▶ **Cloud service:** Facebook, Google Suite, Office365

Distributed Computation

- ▶ **Cloud computing:** Amazon Web Services, Microsoft Azure, Google Search
- ▶ **Cluster computing:** Hadoop, Spark
- ▶ **Distributed database:** Aerospike, Cassandra, Couchbase, Druid

Distributed System Architecture

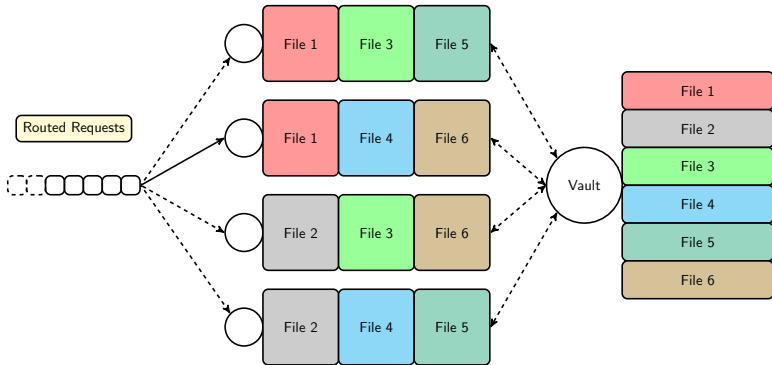
Classification

- ▶ **Client-server:** Online banking, Web servers, e-commerce
- ▶ **Peer-to-peer:** Bitcoin, OS distribution
- ▶ **Hybrid:** Spotify, content delivery in ISPs

Interaction

- ▶ **Master-slave:** Message passing with local memory
- ▶ **Database-centric:** Relation database for interaction

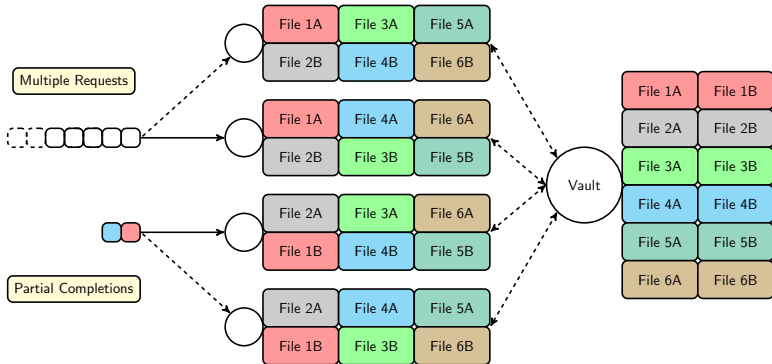
Content Delivery Network



Redundancy for resilience

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

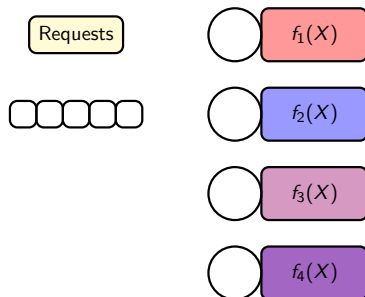
Load Balancing through File Fragmentation



Shared Coherent Access

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

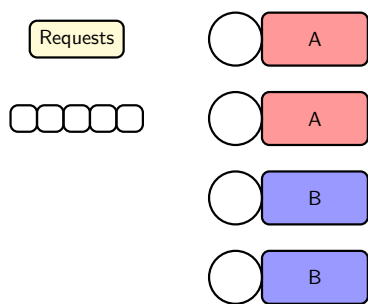
Problem Statement



Compute mean access time to download single message m

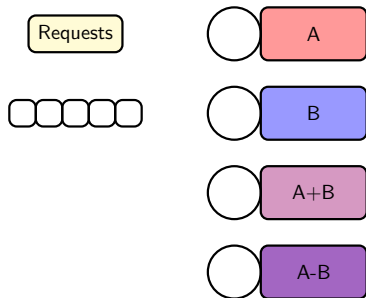
- ▶ with number of **fragments** k such that $m = (m_1, \dots, m_k)$
- ▶ with **encoding** $(f_1(m), \dots, f_n(m))$, and $f_i(m)$ stored at node i

Symmetric Codes



Replication (n, k)

Piece i stored at n/k servers



MDS (n, k)

Whole message can be decoded by any k out of n servers

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

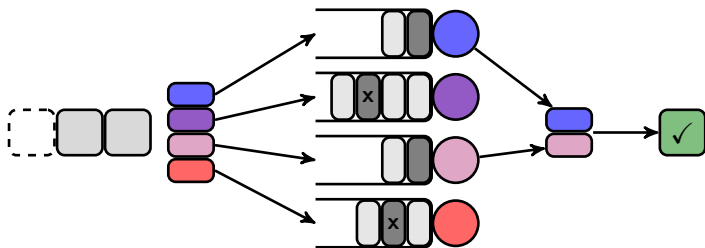
Time in the system

- ▶ Till the reception of whole file

Service at each server

- ▶ IID exponential service time with rate $\mu = k/n$

Storage Coding – (n, k) Fork-Join Model



exempli gratia: Joshi, Liu, Soljanin (2012, 2014), Joshi, Soljanin, Wornell (2015), Sun, Zheng, Koksal, Kim, Shroff (2015), Kadhe, Soljanin, Sprintson (2016), Li, Ramamoorthy, Srikant (2016)

Prior Work and Contributions

Kannan et al: join k queues for replication and MDS codes

- ▶ Numerical bounds using block Markov chains
- ▶ Trade-off between numerical accuracy and computational effort

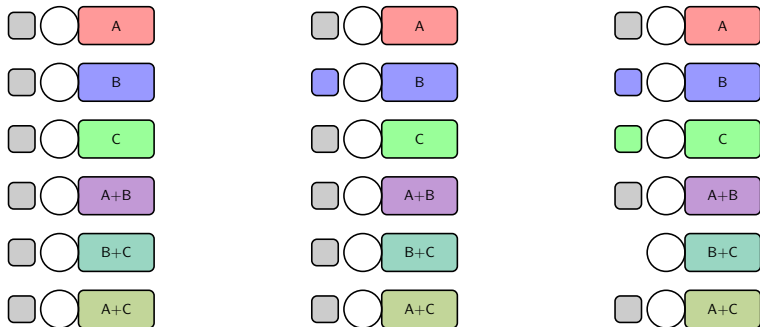
Soljanin, Wornell et al: fork-join (n, k) queues for MDS codes

- ▶ Closed-form upper and lower bounds
- ▶ Loose bounds for most of the rate region

This work: fork-join (n, k) queues for all symmetric codes

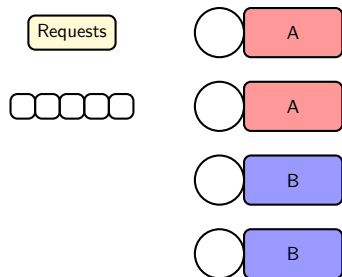
- ▶ Tight closed-form approximations for all rate regions
- ▶ Stability region for all symmetric codes
- ▶ Delay minimising symmetric code

Coding Model



- ▶ Information sets $\mathcal{I} = \{S \subset [n] : |S| = k, f_S \text{ reconstructs } m\}$
- ▶ 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 $N_{|T|} = |M(T)|$

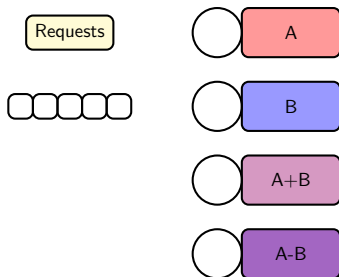
Symmetric Codes



Replication (n, k)

Number of useful servers

$$N_i = (k - i)n/k$$

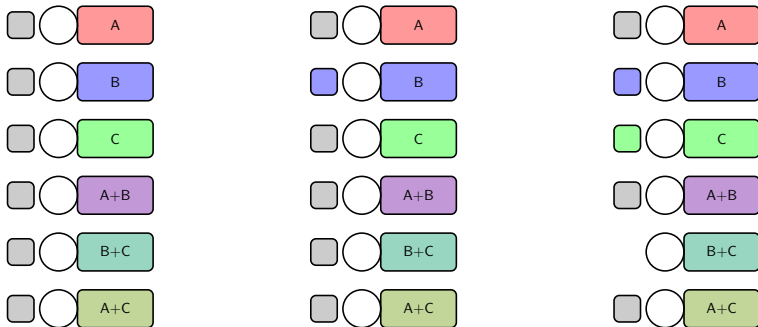


MDS (n, k)

Number of useful servers

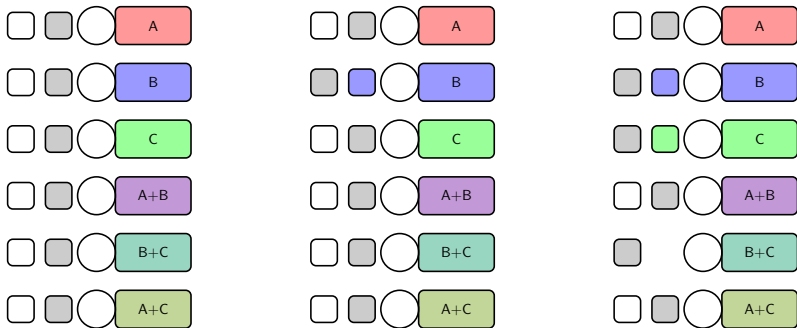
$$N_i = (n - i)$$

Single Request



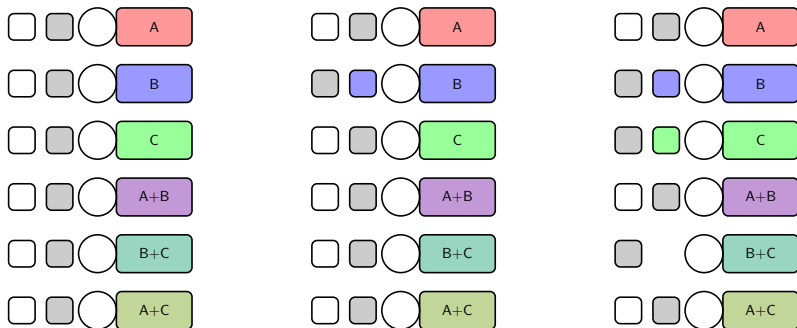
► $\mathbf{T}(t) = \{T \subset S : S \in \mathcal{I}\}$ is a Markov process

Two Requests



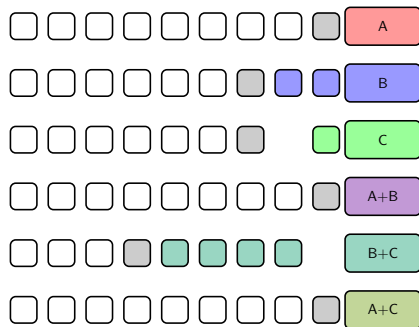
- ▶ $\mathbf{T}(t) = \{(T_1, T_2) \in S \times S : S \in \mathcal{I}\}$ is a Markov process
- ▶ $|T_1| \geq |T_2|$ and $M_{T_1} \subset M_{T_2}$
- ▶ FIFO service: number of available servers $M_{T_2} \setminus M_{T_1}$

State Transitions



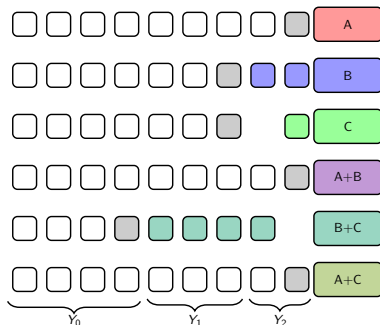
- ▶ Arrival rate: $(T_1, T_2) \rightarrow (T_1, T_2, \emptyset)$ at rate λ
- ▶ Departure rate: $(T_1, T_2) \rightarrow (T_2)$ at rate $N_{|T_1|}\mu$
- ▶ Service rate: $(T_1, T_2) \rightarrow (T_1, T_2 \cup B)$ at rate μ

State Space Collapse



- ▶ $\mathbf{L}(t) = \{(\ell_1, \dots, \ell_r) : \ell_i = |T_i|, \ell_1 \geq \ell_2\}$ is a Markov process
- ▶ **Arrival:** $(\ell_1, \dots, \ell_r) \rightarrow (\ell_1, \dots, \ell_r, 0)$ at rate λ
- ▶ **Departure:** $(\ell_1, \dots, \ell_r) \rightarrow (\ell_2, \dots, \ell_r)$ at rate $N_{\ell_1} \mu$
- ▶ **Service:** $(\dots, \ell_i, \dots) \rightarrow (\dots, \ell_i + 1, \dots)$ at rate $(N_{\ell_i} - N_{\ell_{i+1}}) \mu$

State Space Transformation



- ▶ $\mathbf{Y}(t) = \{Y_0, Y_1, \dots, Y_{k-1}\}$ is a Markov process
- ▶ **Arrival:** $Y_0 \rightarrow Y_0 + 1$ at rate λ
- ▶ **Departure:** $Y_{k-1} \rightarrow Y_{k-1} - 1$ at rate $N_{k-1}\mu$
- ▶ **Service:** $(Y_{i-1}, Y_i) \rightarrow (Y_{i-1} - 1, Y_i + 1)$ at rate $(N_{i-1} - N_{i-1})\mu$

State Transitions of Collapsed System



Arrival of requests at rate λ

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

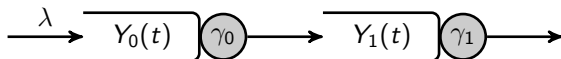
Getting additional symbol at rate $\gamma_i = (N_{i-1} - N_i)\mu$

- ▶ 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 at rate $\gamma_{k-1} = N_{k-1}\mu$

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

Tandem Queue Interpretation (No Empty States)



Duplication

- ▶ n/k available servers at level i
- ▶ Normalized service rate at level i

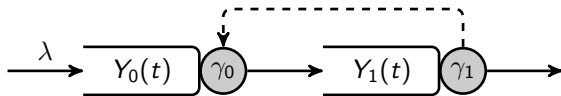
$$\gamma_i = 1$$

MDS Coding

- ▶ Single server 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)



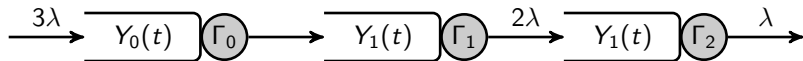
Tandem Queue with Pooled Resources

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

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

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

Stability Region For Pooled Tandem Queues

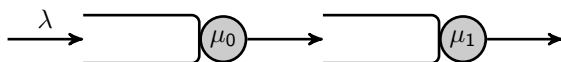


For a distributed storage system with symmetric codes and fork-join queues with FCFS service, the stability region is equal to

$$\lambda < \min \left\{ \frac{\Gamma_i}{k-i} : i \in \{0, \dots, k-1\} \right\},$$

where $\Gamma_i \triangleq \sum_{j=i}^{k-1} \gamma_j$ is the useful service rate for level i .

Bounding and Separating



Theorem[†]

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

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

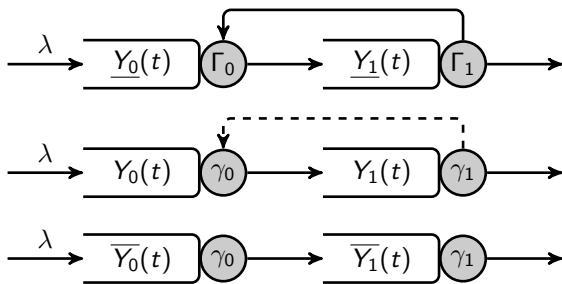
Uniform Bounds on Service Rate

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

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

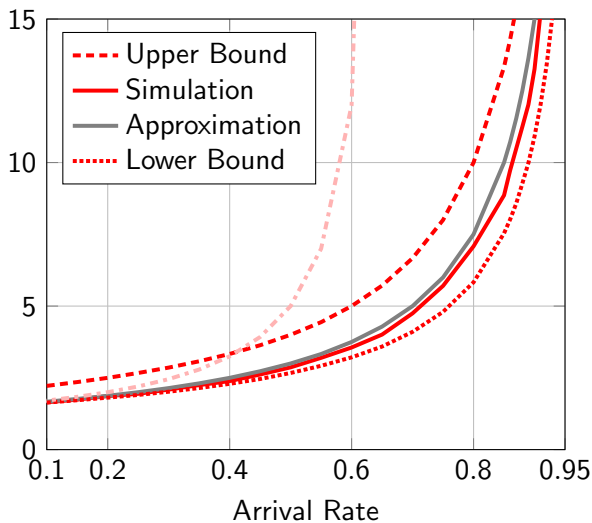
Upper Bound

Lower values for service rate yield upper bound on queue distribution

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

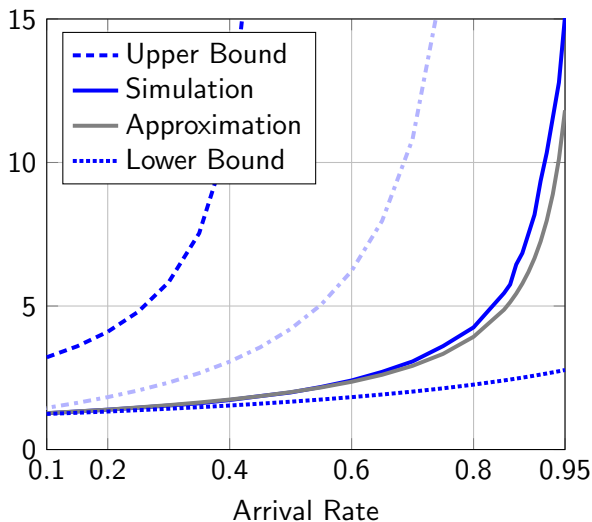
Mean Sojourn Time

Replication Coding

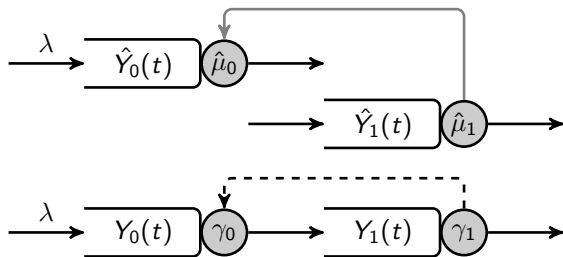


Mean Sojourn Time

(4, 2) MDS Code



Approximating Pooled Tandem Queue



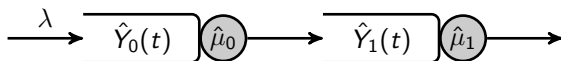
Independence Approximation with Statistical Averaging

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

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

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

Delay Minimizing Storage Code

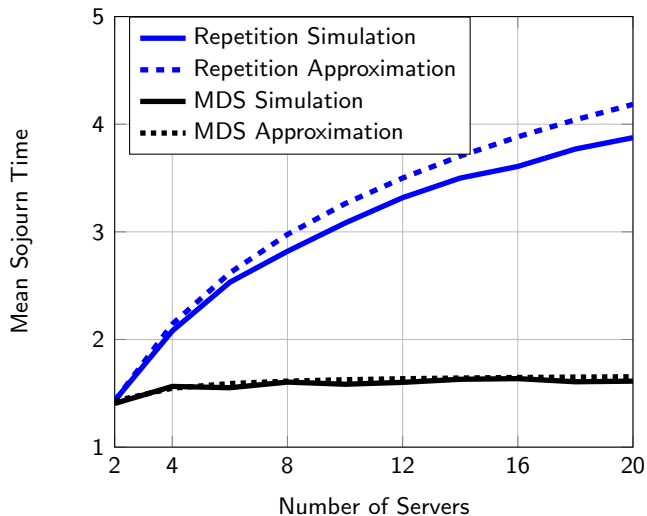


Optimizer to the objective function

$$\gamma^* = \arg \min \left\{ \sum_{i=1}^{k-1} \frac{1}{\Gamma_i - (k-i)\lambda} : \gamma \in \mathcal{A} \right\}.$$

The MDS coding scheme minimizes the approximate mean sojourn time for a fork-join queueing system with identical exponential servers among all symmetric codes.

Comparing Repetition 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