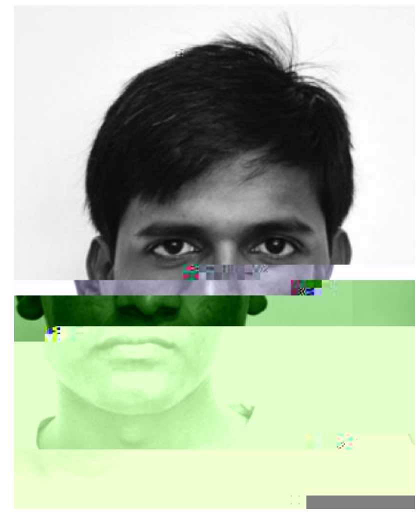
Vaneet Aggarwal, Tian Lan, Parimal Parag

OUR TEAM

Vaneet Aggarwal
Purdue

Tian Lan
GWU



Parimal Parag
IISc

IISc

PURDUE

CLOUD STORAGE

Source: https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2F%40spirosx%2Fthe-beginners-guide-to-the-cloud-

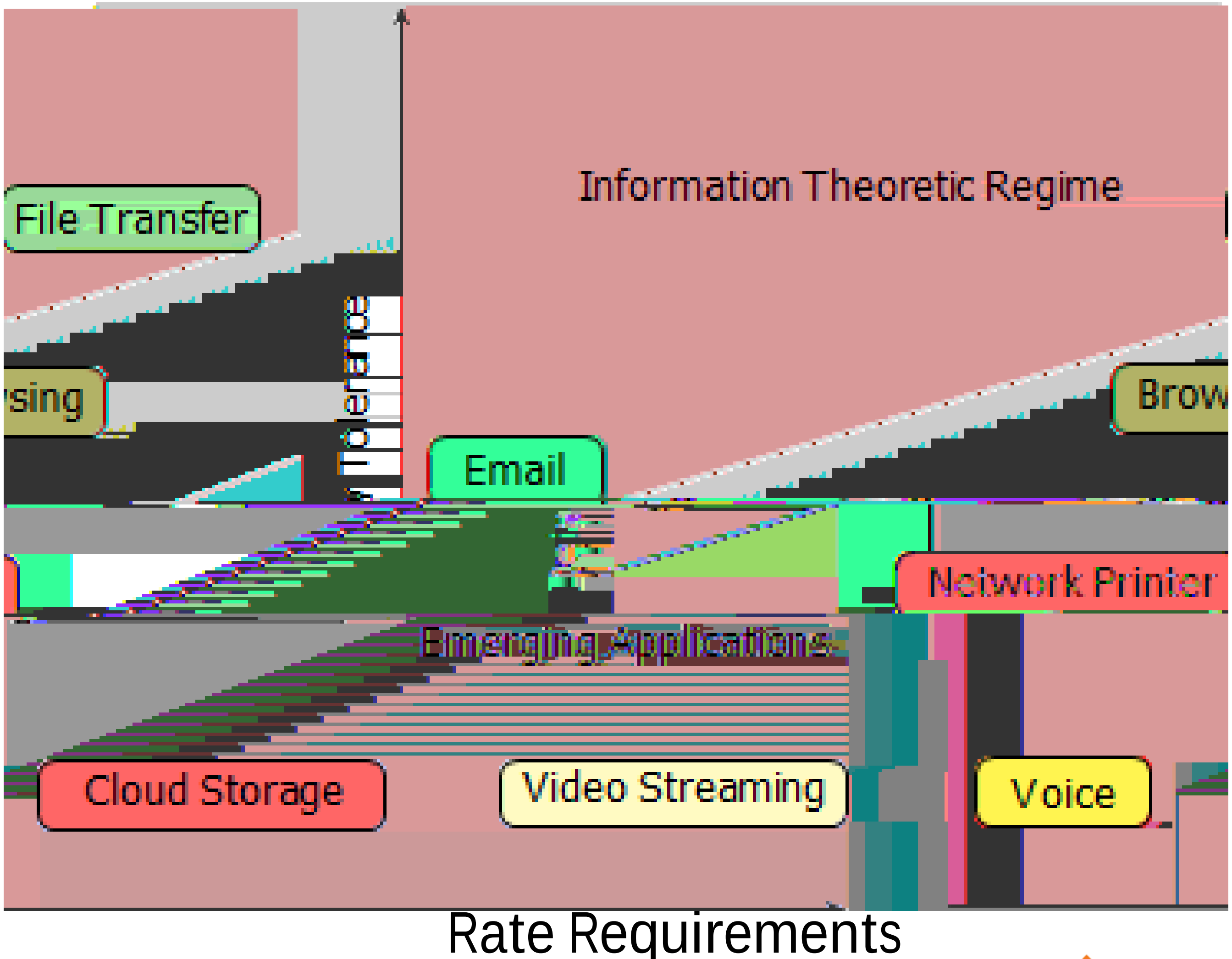
- Demand for storage services increasing rapidly (Backup, photos, videos)
- Companies increasing storage space rapidly (Baidu 2TB, Qihoo 360 36TB)

GROWTH IN CLOUD STORAGE

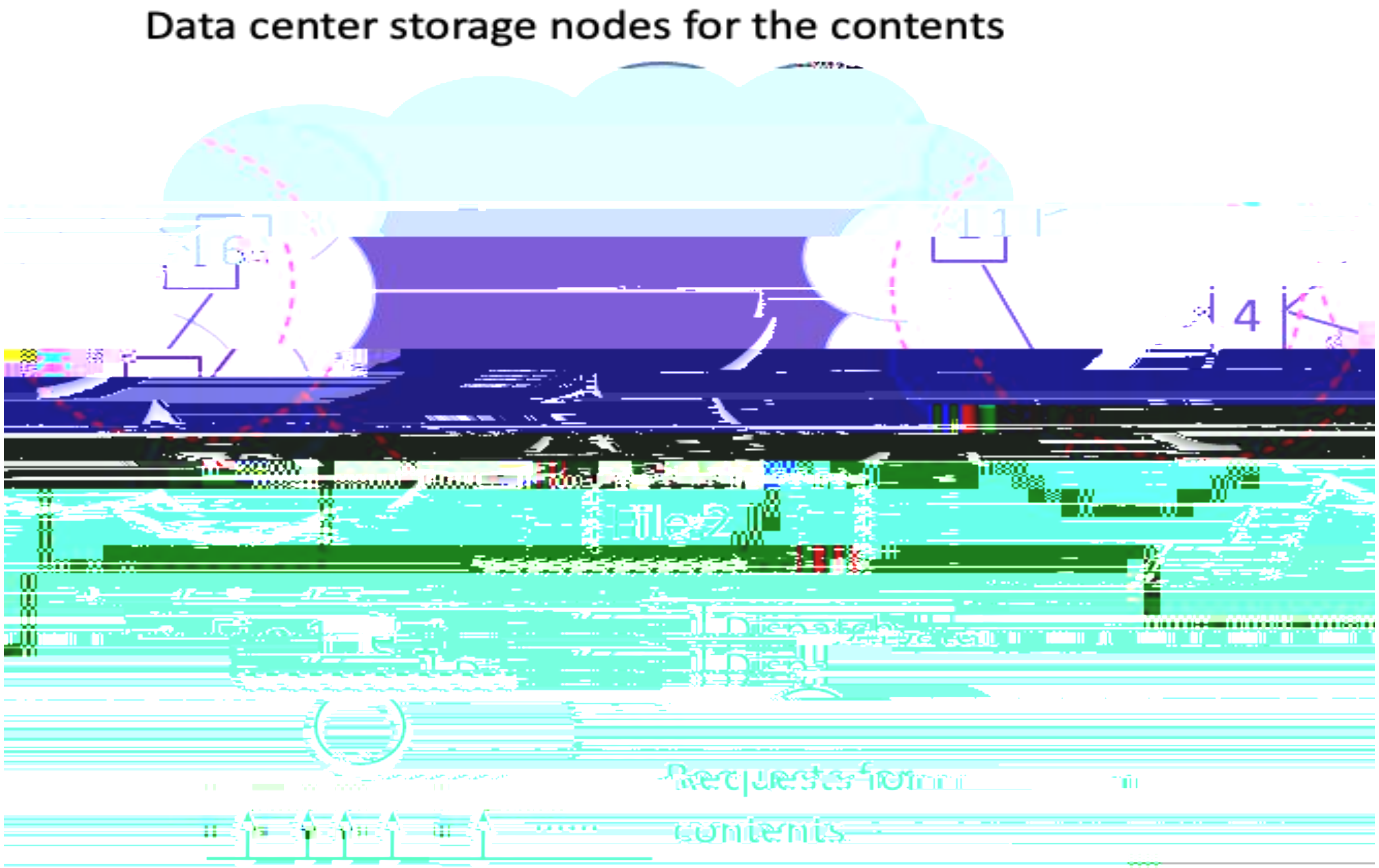
- Growth in personal cloud storage and sharing of photos/videos/documents

- The global cloud storage market size is projected to grow from USD 50.1 billion in 2020 to USD 137.3 billion by 2025, at a Compound Annual Growth Rate (CAGR) of 22.3% during the forecast period (MarketsandMarkets.com).

EVOLVING DIGITAL LANDSCAPE



KEY PROBLEM IN THIS TUTORIAL

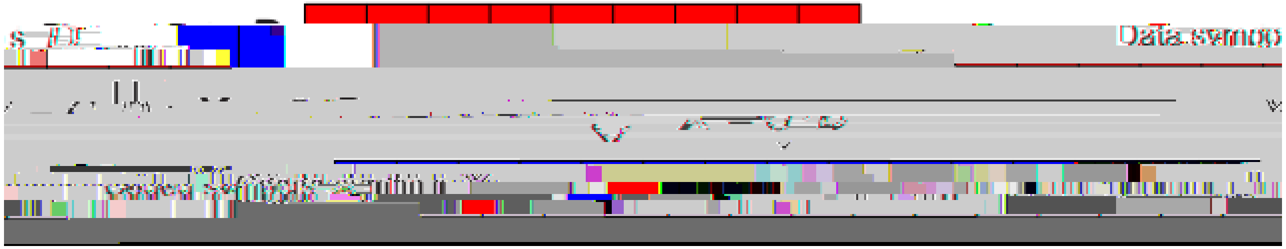


- Modeling, characterization, and optimization of latency for distributed storage systems

PURDUE

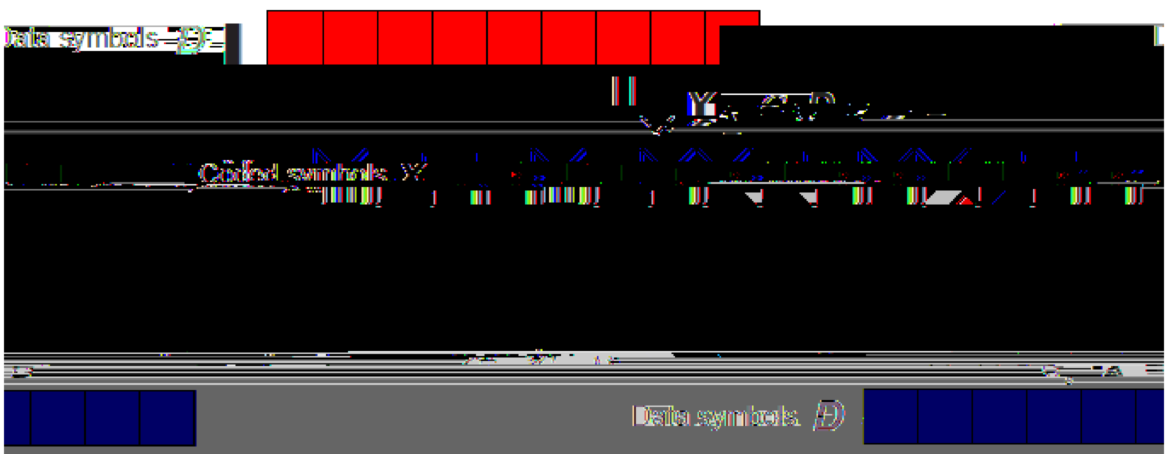
WHAT IS AN ERASURE CODE?

- Erasure Code (EC) involves encoding the message in a redundant manner
- EC transforms message of k symbols to n symbols



WHAT IS AN ERASURE CODE?

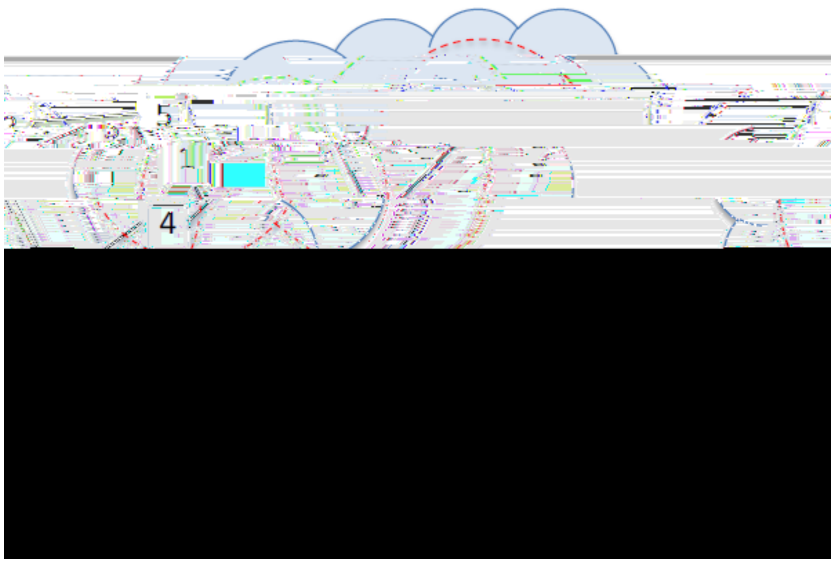
- Erasure Code (EC) involves encoding the message in a redundant manner
- EC transforms message of k symbols to n symbols
- There exists a set of k un-erased symbols for recovery
- For MDS codes, any k un-erased symbols suffice (e.g., Reed-Solomon codes)



- Key Questions:
  - What is the choice of scheduling strategy?
  - How to characterize different measures of latency?
  - How much redundancy to add?
  - What is the optimal placement for coded chunks?

PURDUE

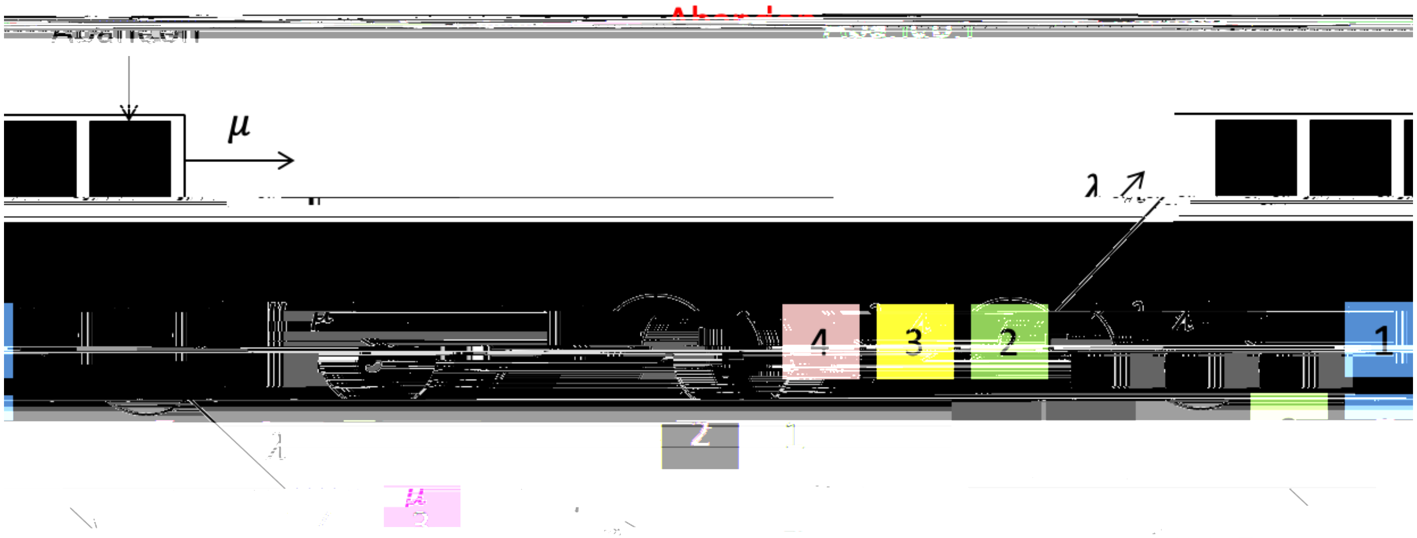
OPTIMAL SCHEDULING IS HARD



Erasure-coded storage.

Scheduling problem.

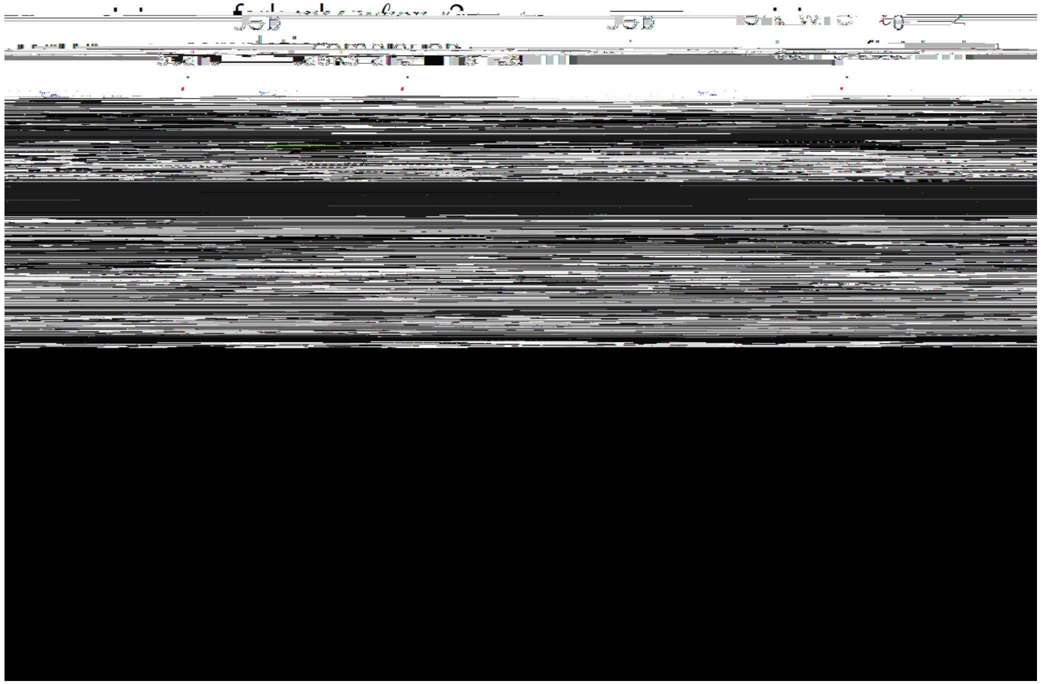
FORK-JOIN SCHEDULING



- (n,k) fork-join queue:
  - Joshi, Liu, Soljanin (2014), Joshi, Soljanin, Wornell (2017)

- Probabilistic scheduling chooses different k-subsets with some probability
  - Xiang, Lan, Aggarwal, Chen (2014, 2016), Aggarwal, Fan, Lan (2017), Alabbasi, Aggarwal, Lan (2019), Wang, Harchol-

DELAYED-RELAUNCH SCHEDULING



- Delayed-Relaunch scheduling: Job at some servers are started with a delay based on completion of some tasks.
  - Badita, Parag, Aggarwal (2020, 2021)

COMPARISON OF STATE-

I2 T2001 P)003 TATE

COMPARISON OF STATE-OF-ART: ANALYSIS RESULTS

[redacted]

OUTLINE

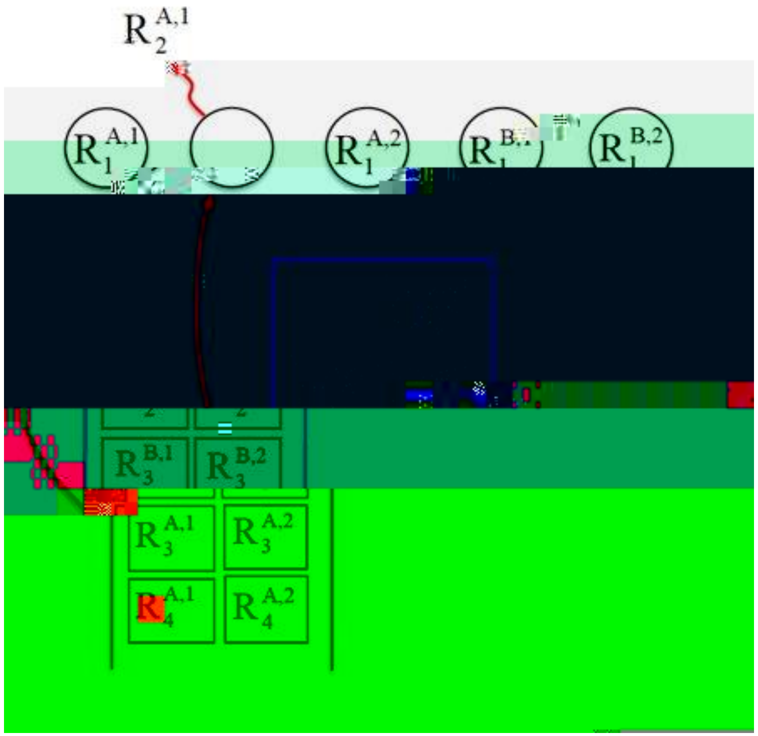
- Introduction
- Fork-join scheduling
- Probabilistic scheduling
- Delayed-Relaunch scheduling
- Evaluations and other applications

- AT&T Research: Yih-Farn Robin Chen (now retired), Moo-Ryong Ra (now at Amazon), Vinay Vaishampayan (now at City University of NY), Chao Tian (now at Texas A&M University)
- Purdue University: Abubakr Al-Abbasi (now at Qualcomm), Jingxian Fan (now at Google), and Ciyuan Zhang
- George Washington University: Yu Xiang (now at AT&T)
- IISc Bangalore: Ajay Badita (now at IOTA), Rooji Jinan, Saraswathy Ramanathan, Vikram Srinivasan
- IIT Madras: Pradeep Sarvepalli
- Rutgers University: Rawad Bitar (now at TUM), Salim El Rouayheb
- Texas A&M University: Jean-Francois Chamberland
- University of Illinois, Chicago: Balajee Vamanan

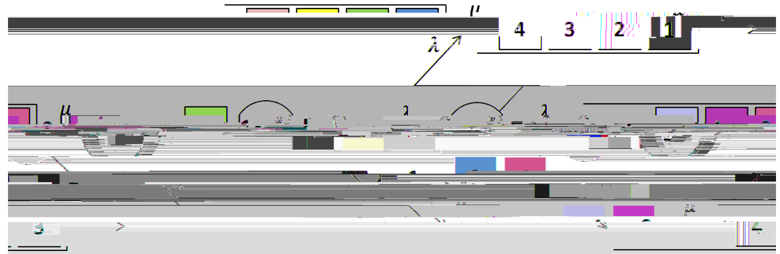
- Funding:
  - NSF CNS 161860061035 720 540 refQ EMC /Art1f3X2a0ang (en-US)>BDC q0.000010729 0 720 540h51*nBT/F5 11.

STORAGE BOOK

- Promotion Code: 994513



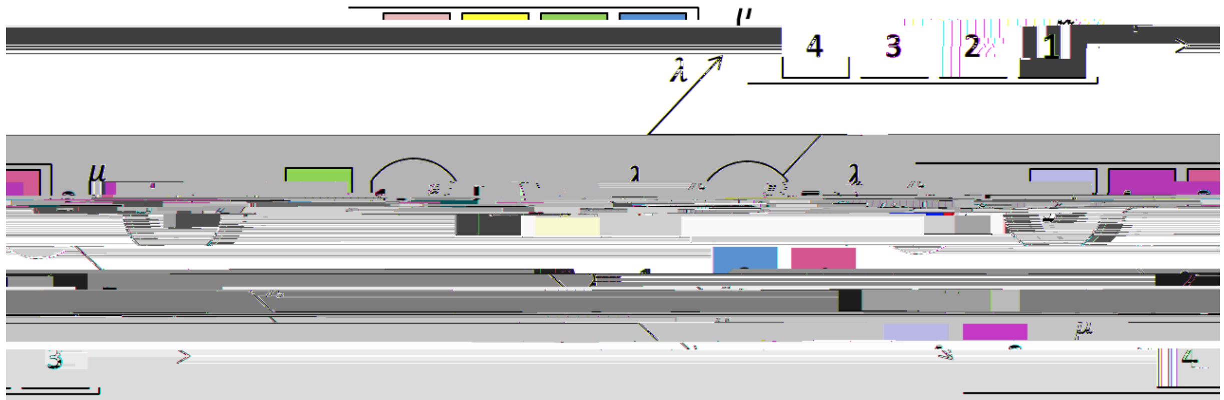
Scheduling problem in erasure-coded storage.



For the (n,k) fork-join system to be stable, the Poisson arrival rate λ and the service rate μ per server must satisfy $\lambda < n\mu/k$

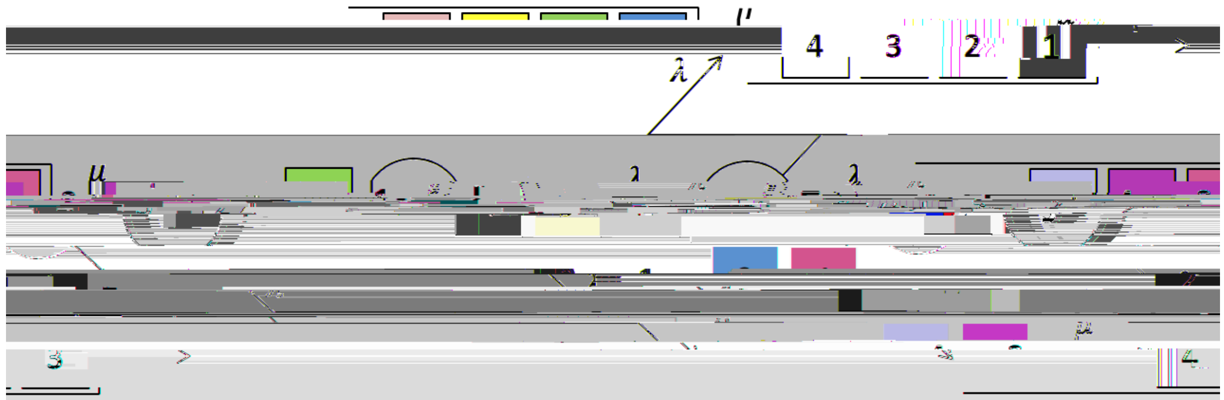
- Proof outline:
  - When k out of the n tasks finish service, the remaining $n-k$ tasks abandon their queues
  - A task can be one of the abandoning tasks with probability $(n-k)/n$.
  - The effective arrival rate to each queue is λ minus abandonment $\lambda(n-k)/n$.
  - $\lambda - \lambda(n-k)/n < \mu$ gives the condition.

Fork-join:

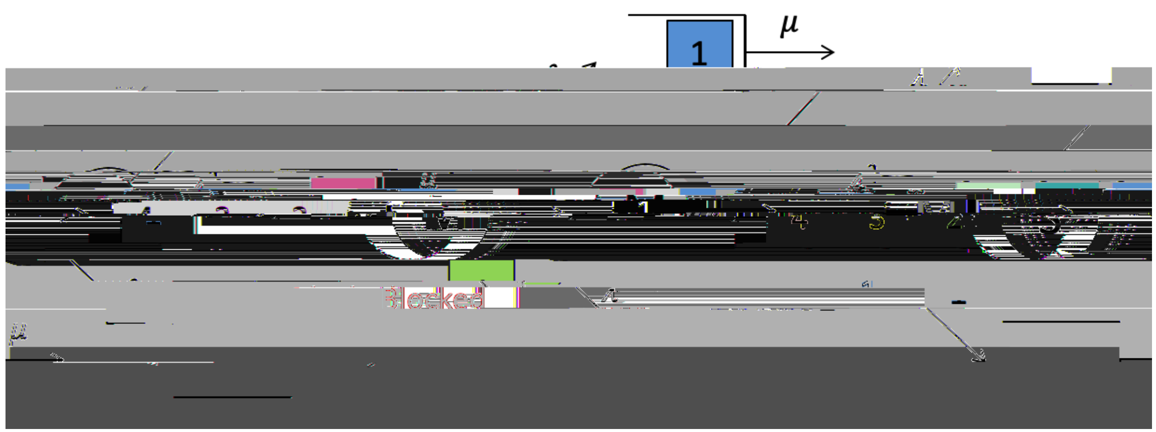


- Recall: Latency is defined as the average time spent in the fork-join system.
- Analyzing the waiting time using Markov Chains requires:
  - Modeling individual queue evolutions that are dependent
  - Encapsulating the execution history in MC

Fork-join:

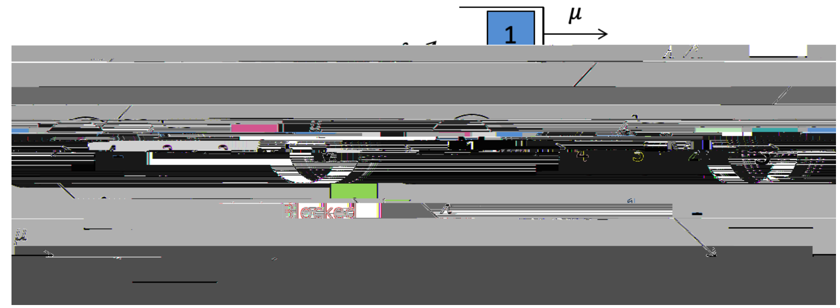


Split-merge:



Split-merge queues provide an upper bound on fork-join.

- (n,k) split-merge is equivalent to an $M/G/1$ queue.
  - Arrivals are Poison with rate λ.
  - Service time X is the kth order statistic.



- Find $E[X]$ and $Var[X]$.
  - Independent services times at the servers.
  - Analyze the kth order statistic of exponential distributions of $Exp(\mu)$.
- Compute the average latency:
  - Use the Pollaczek-Khinchin formula for $M/G/1$ queue.
- It gives an upper bound on the latency of fork-join system.

- Given i.i.d. service times $X_1, X_2, \ldots, X_n$.
- Equivalent service time $Y = X_{(k)}$, i.e., the kth smallest of $X_1, X_2, \ldots, X_n$
- Distribution for kth order statistic:
  - [illegible]

- The Pollaczek-Khinchin formula for $M/G/1$ queue with service time X:

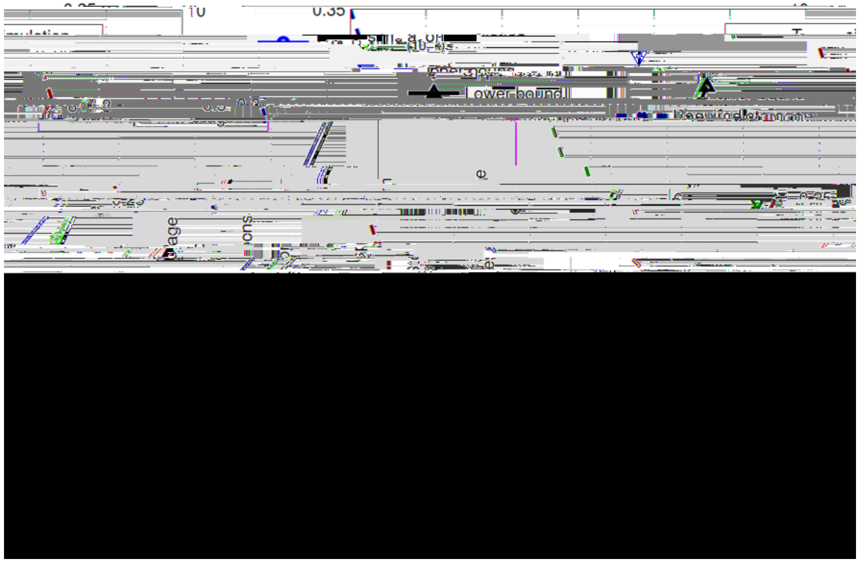
$$E[T] = E[X] + \frac{\lambda E[X^2]}{2(1-\lambda E[X])}$$

- Substituting the values of $E[X]$ and $Var[X]$, we find an upper bound on the latency of fork-join systems.

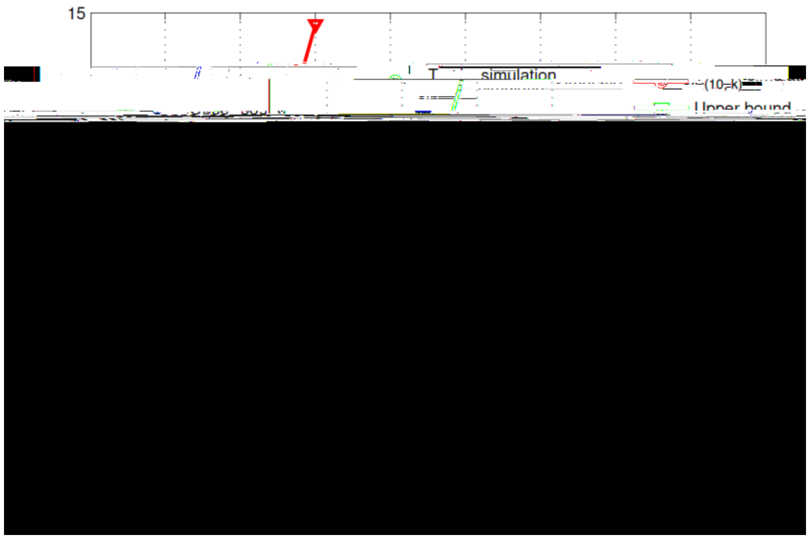
PURDUE

PURDUE

The expected latency "Y ححڋچ *for an*



Arrival rate _=1 and service rate ‘ =10.



Arrival rate _=1 and service rate ‘ =1.25.

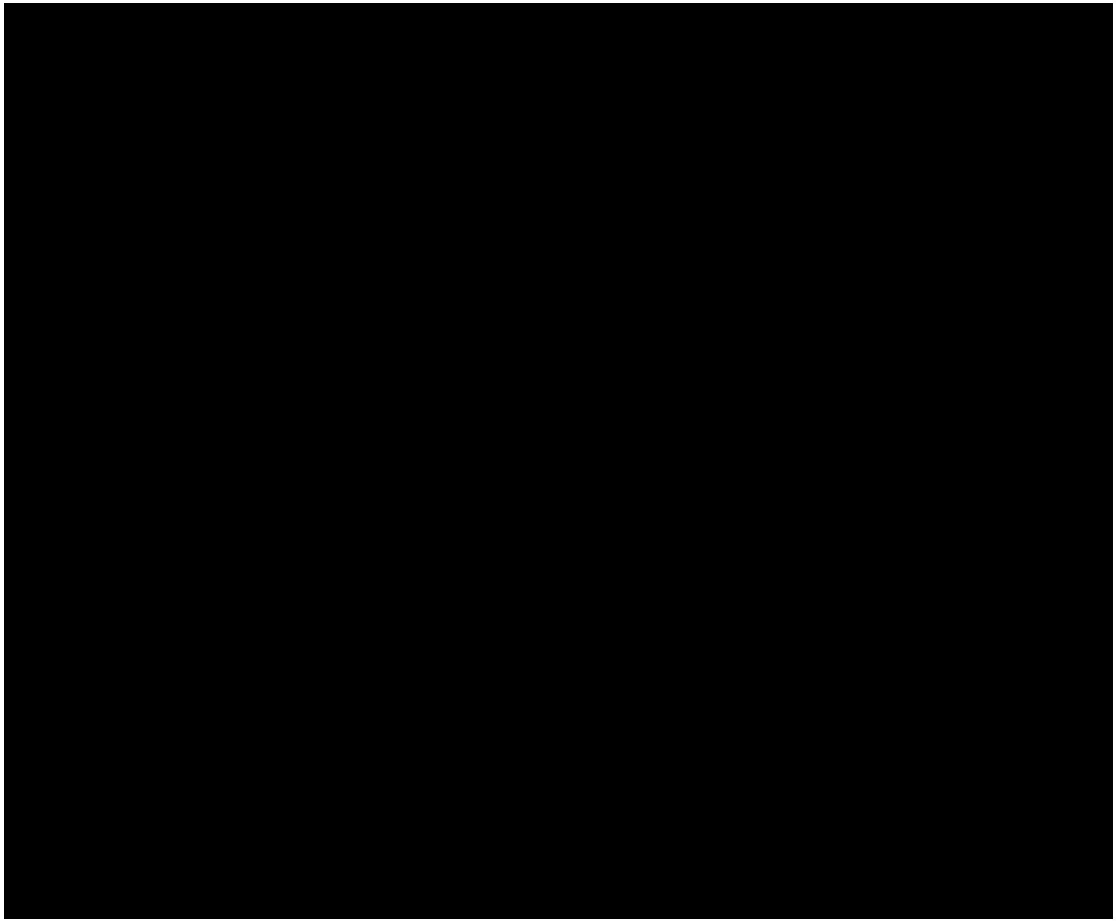
[1] Joshi, Soljanin, Wornell (2015).

PURDUE

The expected latency ῎Ɣ ☐ʻ♮ɕ for an (ὲ♮῀Ώ) fork-join system can be approximated by:

῎Ɣ ☐ʻ♮ɕ ... ρ / ((ὲ ῀Ώ) ‘ ῀Ώ ῀Ώ ═) ꟹ

- The lower bound is valid only when ῀Ώ═ ὲ ‘ ꟹ
- This stability condition is the same as that of fork-join systems.



We choose arrival rate _ π8σ and service rate ' ΩT è π8υ.

- i.i.d. and general service times:
- Each file i encoded using an (n_i, k_i) code and has arrival rate λ_i :

- Fork-join systems provide an analytical framework for the study of erasure-coded storage, e.g.,
  - minimizing file access latency.
  - optimizing coding strategy.
- Upper and lower bounds to analyze the latency of general codes.
- A tight closed-form approximation of average latency.
- Average latency is better for MDS codes for all code

- Tight upper bound:
  - There is still a large gap between the upper bound and the optimal stability conditions even for exponential service times.
- General file placement:
  - When each file is placed on a subset of the servers, no latency result is available for this general setting.
- Heterogeneous servers:
  - Analyzing the latency for heterogeneous servers with different service time distribution is still an open problem.
- Approximations and guarantees:
  - In the asymptotic regime?

UNIVERSITY

$$\sum^{m}$$ [illegible]

$$\sum^{m}$$

$$\sum^{m}$$ [illegible]

$$\sum^{m} \text{[illegible]}$$

UNIVERSITY

UNIVERSITY

UNIVERSITY

UNIVERSITY

	[illegible] Queries/User	Query Refinement	Revenue/User	Any Clicks	Satisfaction	Time to Click (increase in ms)
50ms	-	-	-	-	-	-
200ms	-	-	-	-0.3%	-0.4%	500
500ms	-	-0.6%	-1.2%	-1.0%	-0.9%	1200
1000ms	-0.7%	-0.9%	-2.8%	-1.9%	-1.6%	[illegible]
2000ms	-1.8%	[illegible]	[illegible]	-4.3%	-4.4%	-3.8%

[illegible]

[illegible]

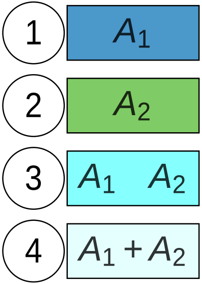
[illegible]

$$\Pr(L_i > x) = \begin{cases} (x_m/x)^\alpha & x \geq x_m \\ 0 & x < x_m \end{cases}$$

$$\Pr(L_i > x) = \begin{cases} (x_m/x)^\alpha & x \geq x_m \\ 0 & x < x_m \end{cases}$$

UNIVERSITY

Coded access model



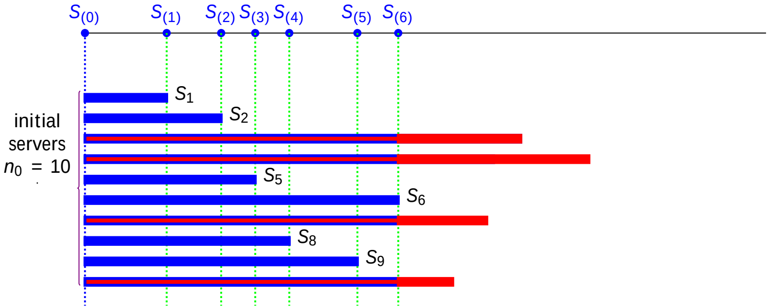
Latency energy tradeo

- Parallelization leads to download speedup
- Redundancy leads to increased energy consumption

Coded access model

Coded access model

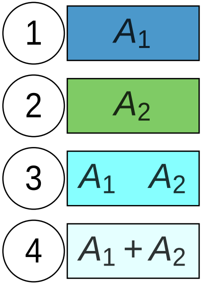
c-shifted unit-rate exponential download times



Download times $(S_1, \ldots, S_n)$

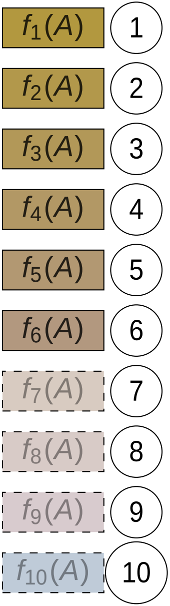
To code or not code?

Shifted exponential download times



$(n; k$

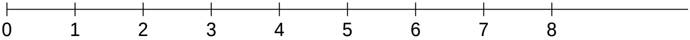
Forking additional servers



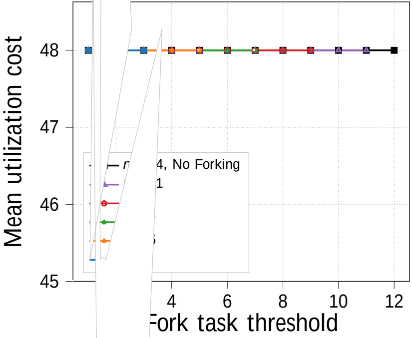
Delayed start of requests in multiple stages

- Stage i starts with download from additional n_i servers
- Stage i ends when downloaded from ℓ_i servers
- Design variables are $(n_i; \ell_i)$ for each stage i

Performance Metric Computation



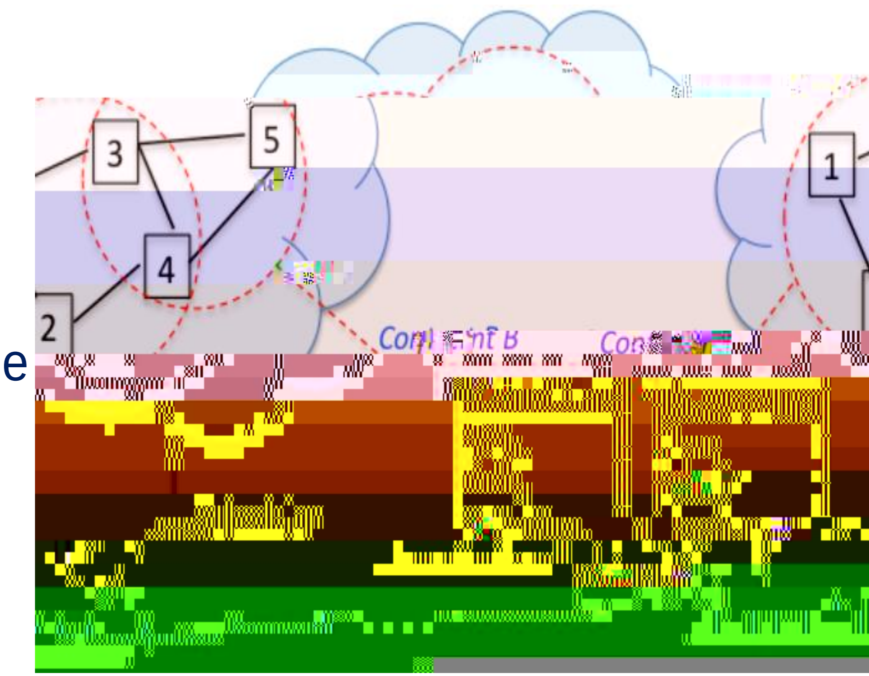
Initial servers n_0 smaller than sub-tasks k



PART 5: EVALUATIONS AND OTHER APPLICATIONS

REQUIREMENTS FOR A DISTRIBUTED STORAGE SYSTEM

- Where to place content?
- What code parameters to choose?
- Which disks to choose for access when the content is requested?

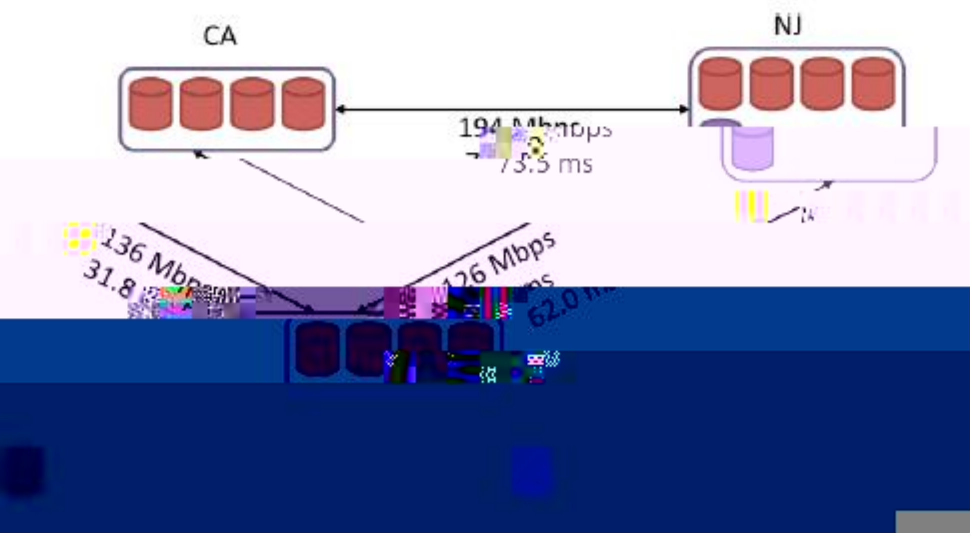


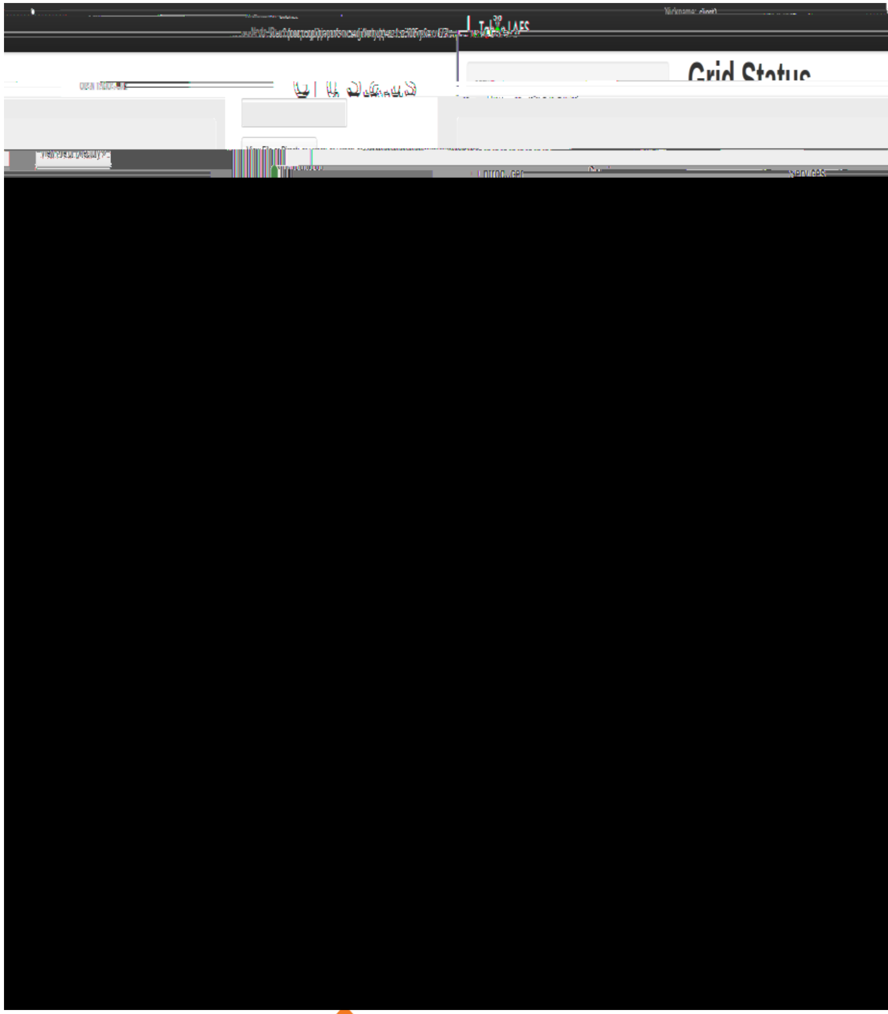
- Baseline:
  - where to place contents: Random
  - what code to use: Fixed
  - from where should content be served: Lowest queue servers

VALIDATION ON OPEN SOURCE STORAGE SYSTEM

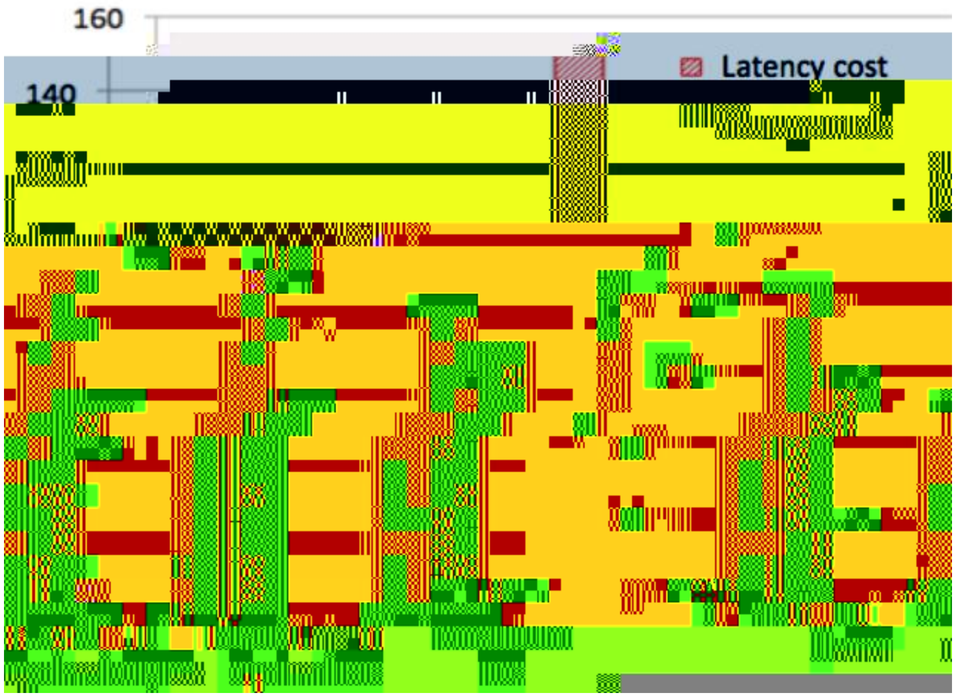
Tahoe-LAFS (Tahoe Least-Authority Filesystem) is an open source, secure, decentralized, fault-tolerant, [illegible] can be used as an [illegible] backup [illegible] peer-to-peer distributed [illegible] and distributed file system.[4][5] It [illegible] system, [illegible] multiple disks to [illegible] access files in the Tahoe system. Tahoe can also be used in a RAID-like manner to use [illegible] make a single large RAID pool of reliable data storage.

SETUP OF STORAGE SERVERS FOR VALIDATION



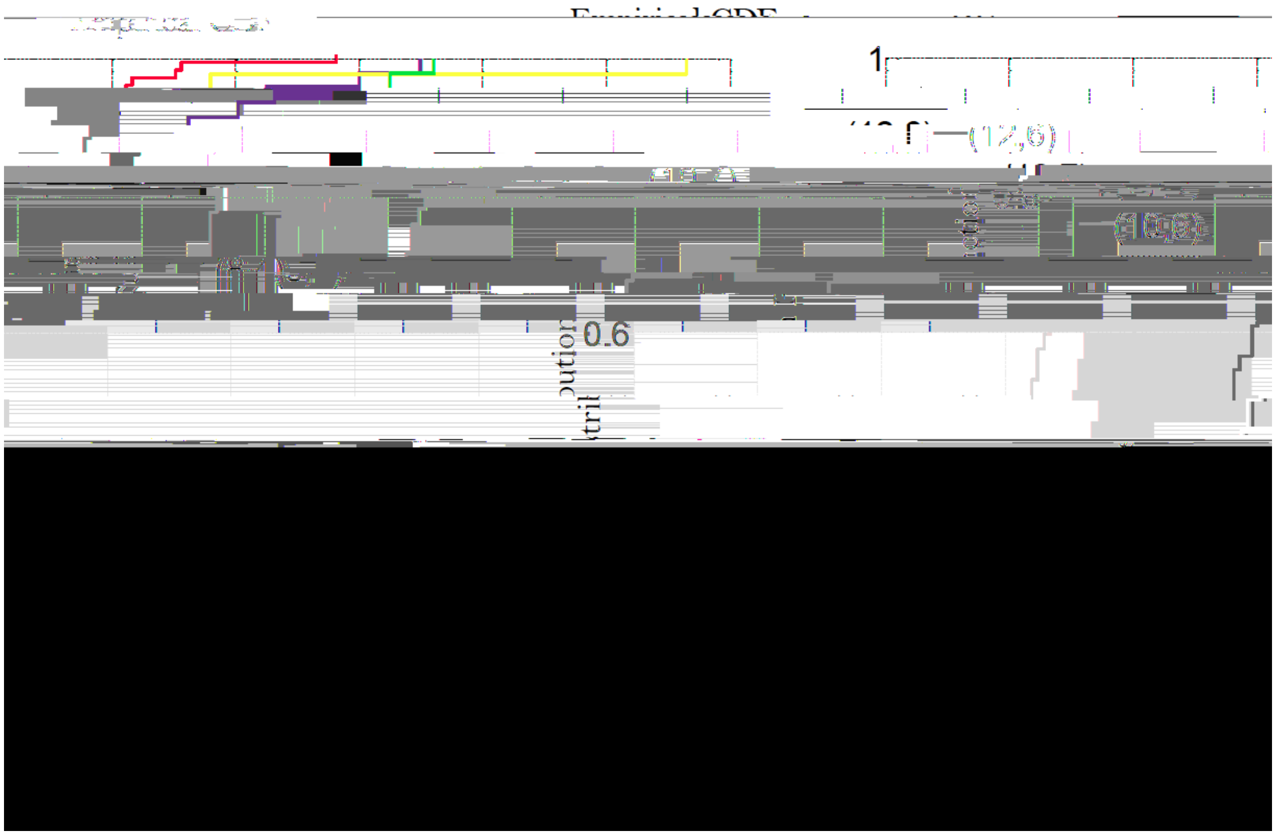


JOINT OPTIMIZATION (CODE, PLACEMENT, ACCESS) IS NEEDED



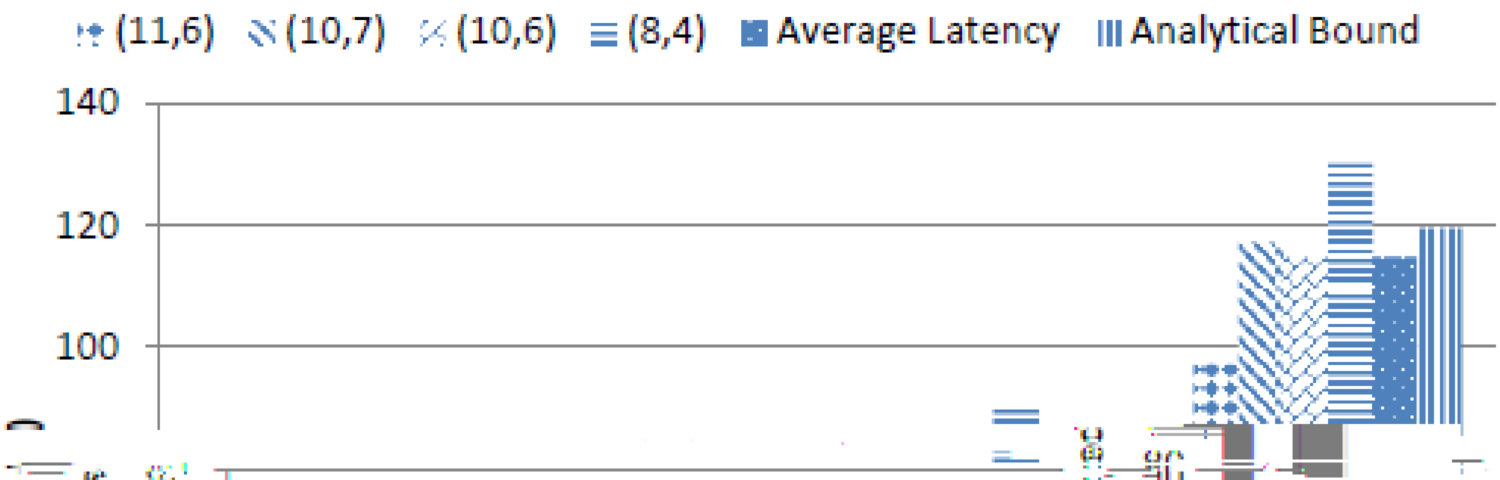
- 1000 files, size 150MB. Cost: $1 for 25MB, tradeoff factor of 200 sec/dollar, chunk size 25MB
- Oblivious LB: Select nodes with probability proportional to service rate
- Random placement: Chooses best outcome of 100 random runs

LATENCY DISTRIBUTION



- 1000 files of size 150 MB, using erasure codes (12, 6), (10, 7), (10, 6), and (8, 4), aggregate rate at 0.118/s.

LATENCY INCREASES SUPER-LINEARLY WITH FILE SIZE



[redacted]

TRADEOFF CURVES

- Visualization of latency and cost tradeoff for file size of (150, 150, 100)MB and arrival rates 1/(30 sec), 1/(30sec), 1/(40 sec).

OTHER APPLICATIONS

- Caching
- Video streaming over Cloud
- Memory-constrained system
- Coded Computing

OTHER APPLICATIONS

- Caching
- Video streaming over Cloud
- Memory-constrained system
- Coded Computing

- Caching is used to reduce

PURDUE

- Erasure Coded Systems allow for functional caching

- Rather than exact chunks, {

LATENCY CALCULATION WITH FUNCTIONAL CACHING

- The latency calculations remain the same as before except that the number of servers to access changes from k to k-d.

- This helps reduce the latency with caching.

- Specific choice of d chunks in the cache will have also change the possibility of accessed servers, while functional caching is more flexible due to using (n+k,k) rather than (n,k) code.

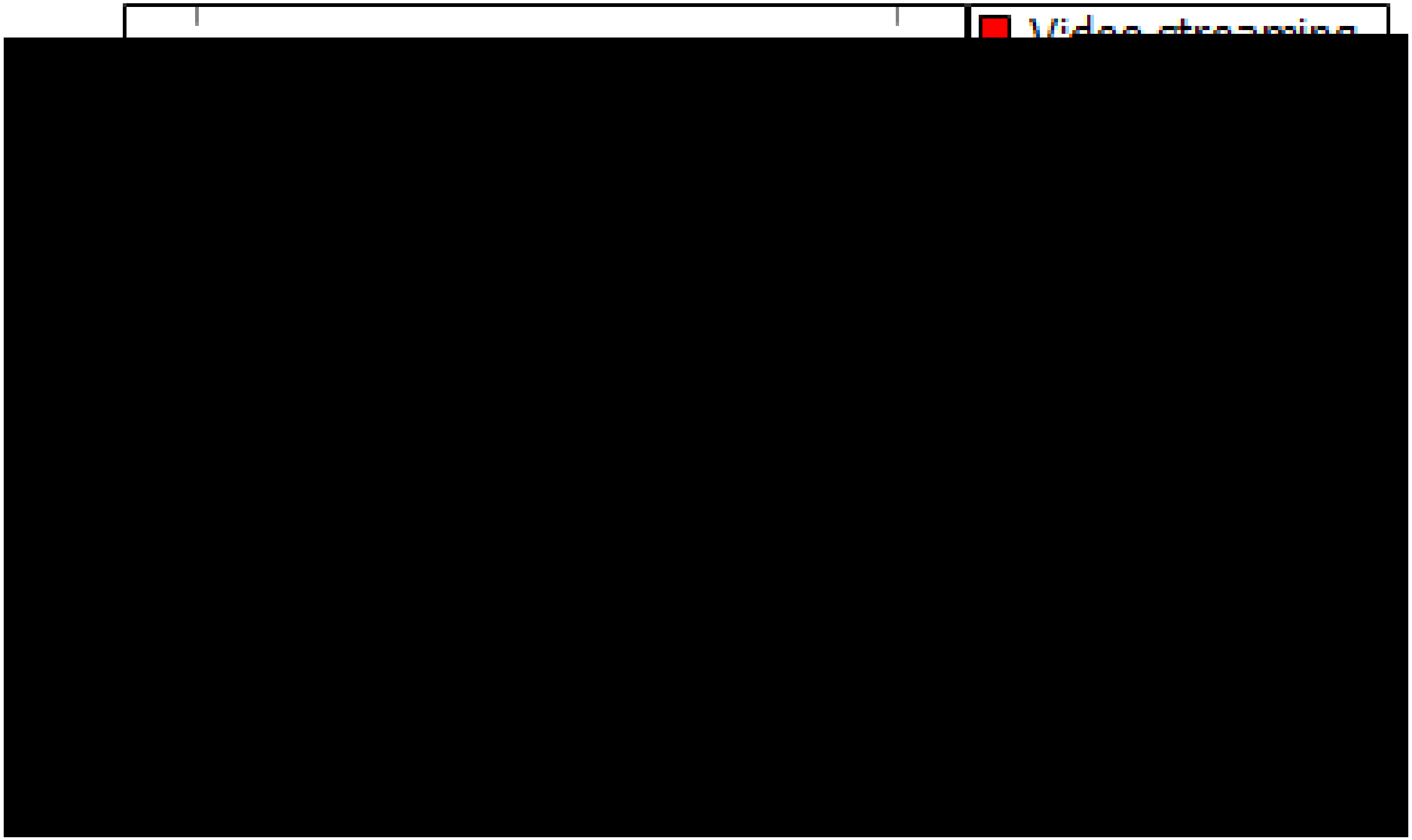
IMPACT OF FUNCTIONAL CACHING

1000 files 100 MB each, (n=7,k=4)

1000 files, (n=7,k=4), cache size 10GB

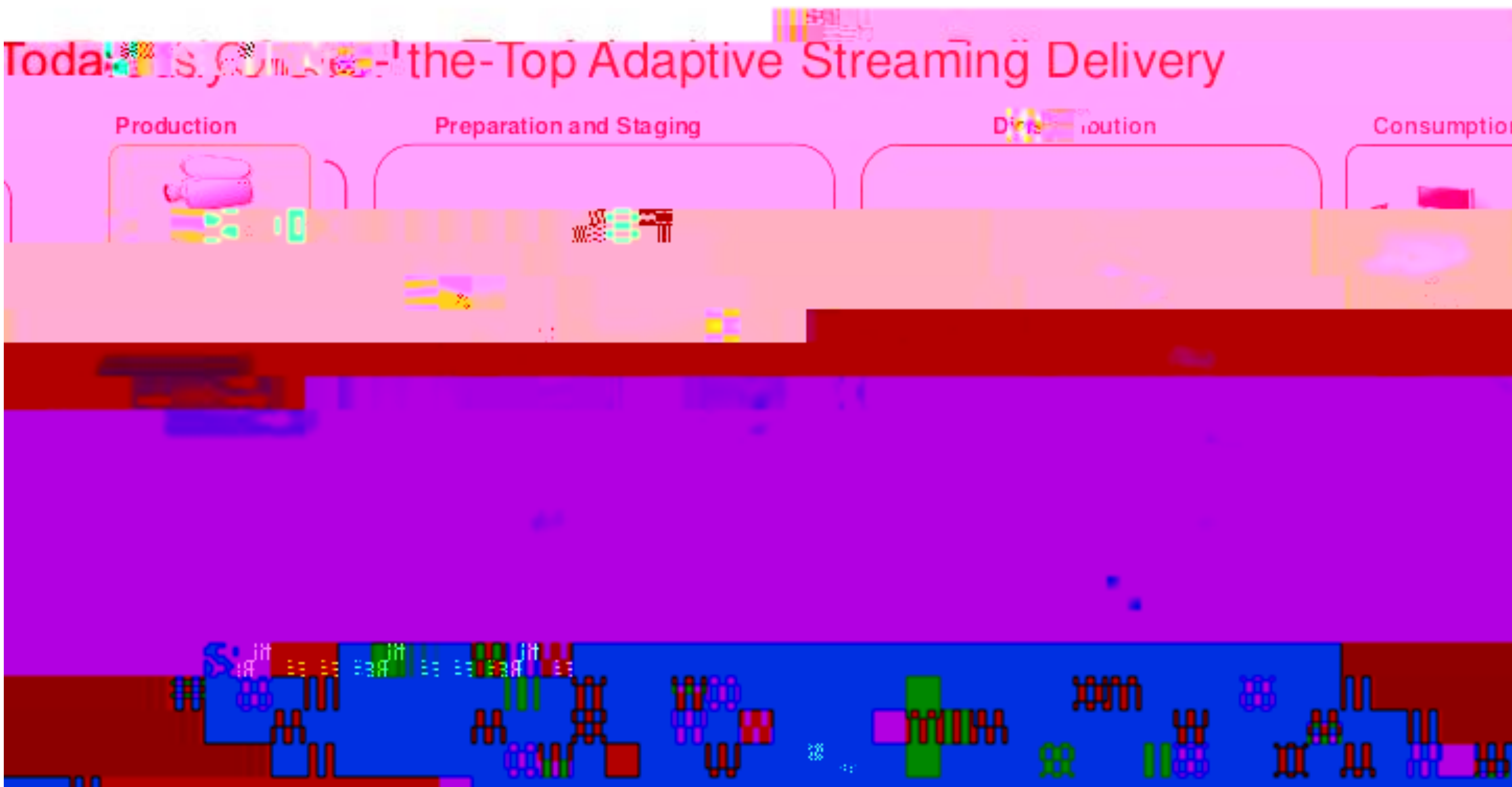
- Caching
-

GOLBAL APPLICATION TRAFFIC SHARE 2021



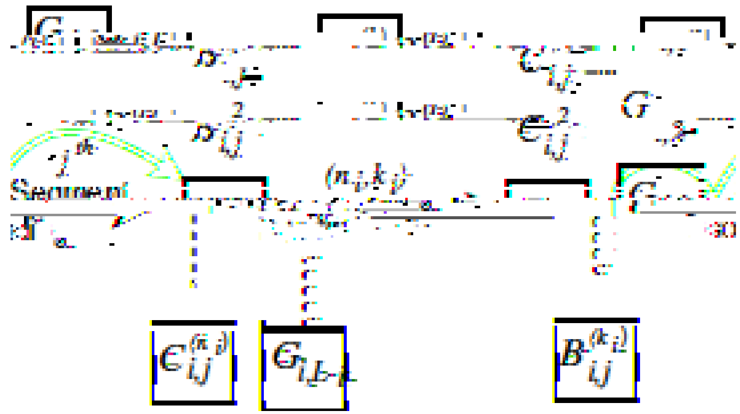
MOTIVATION

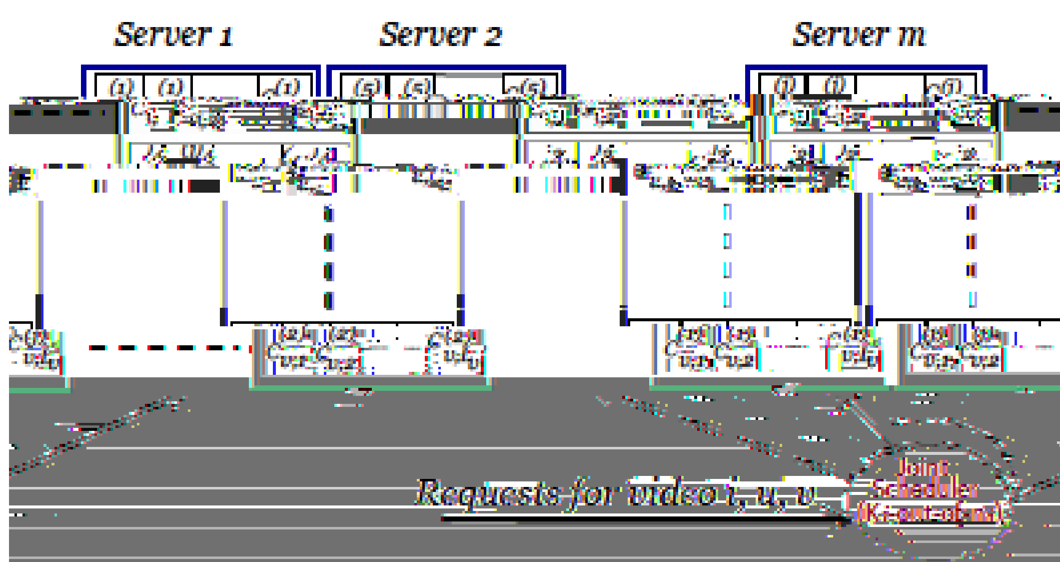
- Video streaming applications represents 62% of the Internet traffic in US
- More than 50% of over-the-top video traffic is now delivered through CDNs



VIDEO STREAMING

- Video Streaming rather than file download.
- Each chunk is erasure-coded
- Coded chunks on server
- Ques: How does servers stream video?





STALL DURATION

- Video Streaming rather than file download.
- Ques: How does servers stream video?
- Approach

[redacted]

- Metric: Stall Duration. Very different from download time since stalls happen anywhere, and all correlated segments need to be accounted.
- Characterized mean and tail of stall durations for this model.

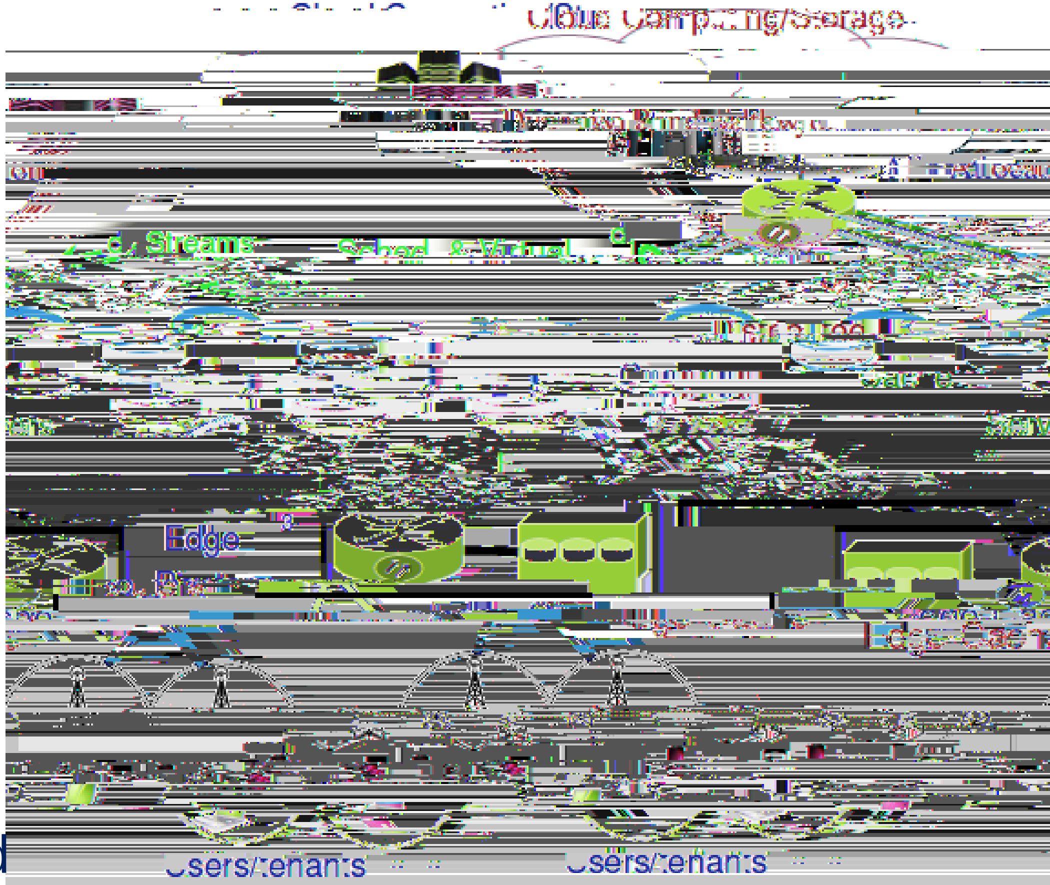
- Compute the time in the queue for each server. Consider the entire data of a file in server j, the requests are still Poisson.

- The start of service with additional of codedD 4/Lang (en-US)>BDC q0.0000105iUI8

BEYOND SINGLE TIER

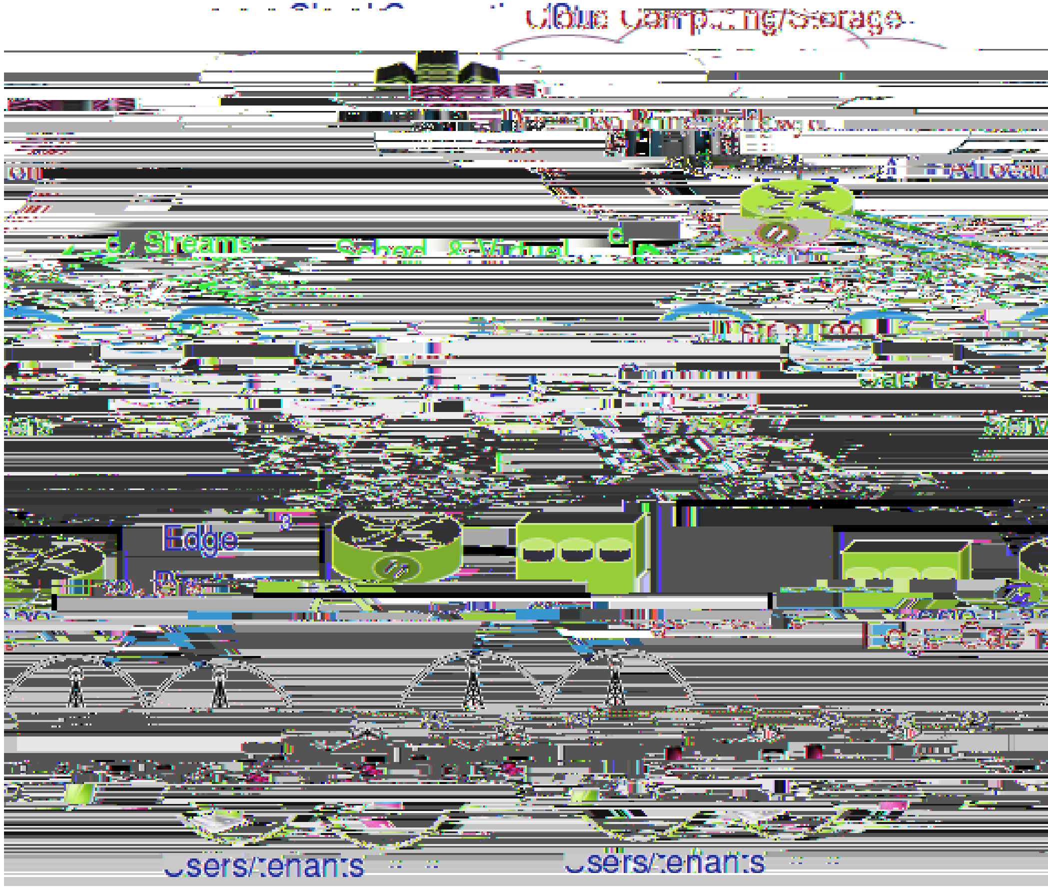
BEYOND SINGLE TIER

- Multiple CDNs
- Caching at CDNs
- Caching in Edge cache
- Edge cache allows for multicast since a later user can get previous content from cache.
- CDN Cache policy: How many initial chunks of each file?
- Edge Cache policy: Each requested file is cached for a certain time, and if not re-requested removed.

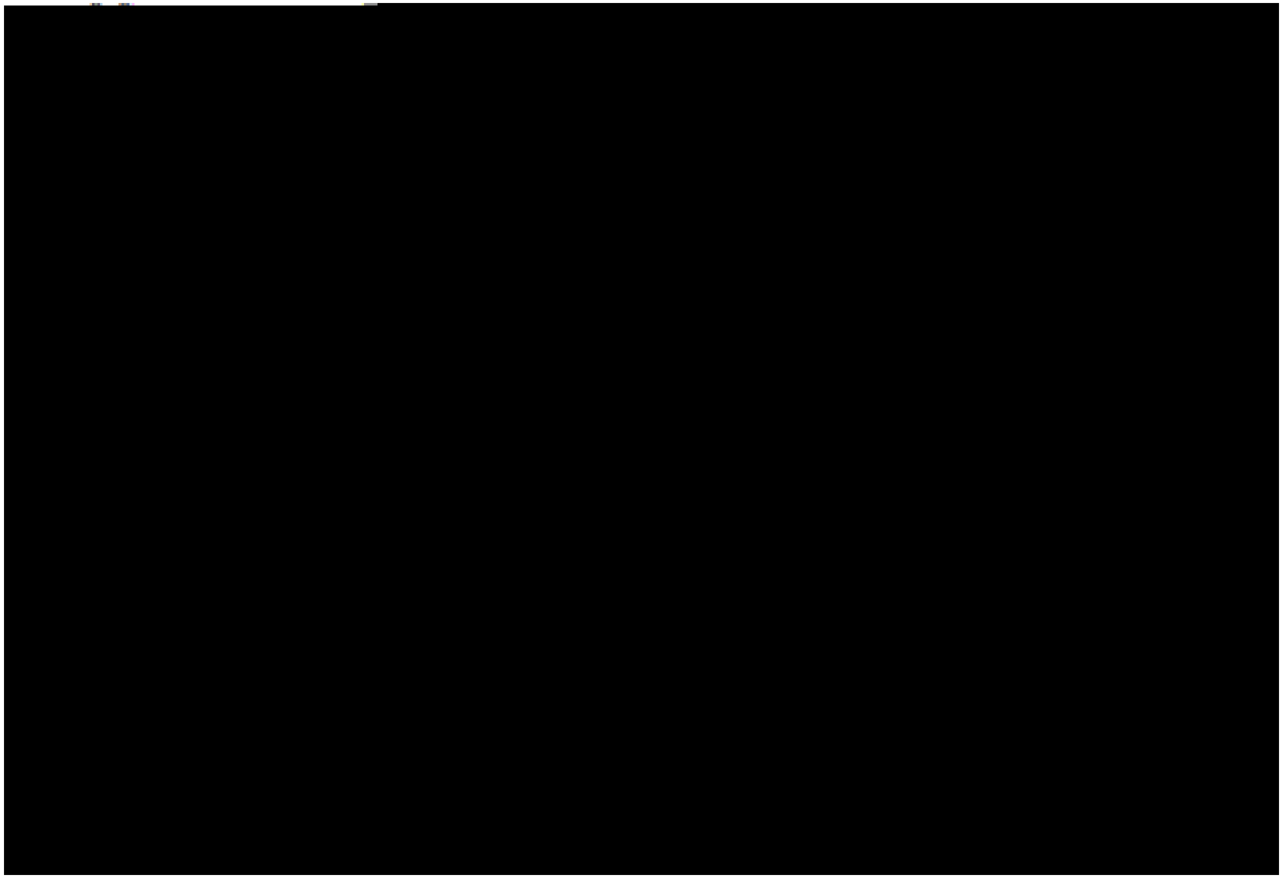


OPTIMIZATION PARAMETERS AND METRIC

- Access probabilities for CDNs, and the different streams from CDN



OPENSTACK IMPLEMENTATION RESULT



- CHF: Caching hot files, PSP: projected proportional service, PEA: Equal probability access, PEC: Projected Equal Caching.

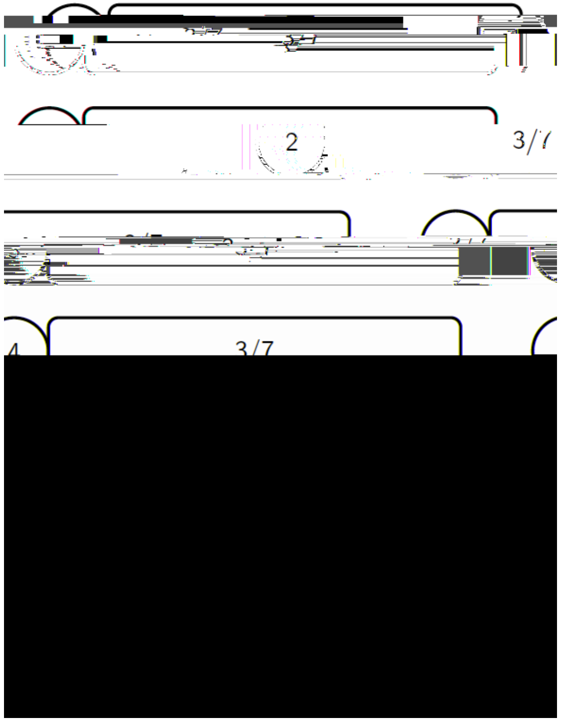
SUMMARY

- New framework for video streaming over CDN
- Gave new bounds for stall duration with multiple flexibilities
- The results demonstrate improved performance metrics
- Single Tier
  - Alabassi and Aggarwal, "Video Streaming in Distributed Erasure-coded Storage Systems: Stall Duration Analysis," IEEE/ACM Transactions on Networking, vol. 26, no. 4, pp. 1921-1932, Aug. 2018.
  - Al-Abbasi and Aggarwal, "VidCloud: Joint Stall and Quality Optimization for Video Streaming over Cloud," ACM Transactions on Modeling and Performance Evaluation of Computing Systems, article no. 17, Jan 2021
- Multi-Tier
  - Alabbasi, Aggarwal, Lan, Xiang, Ra, and Chen, "FastTrack: Minimizing Stalls for CDN-based Over-the-top Video Streaming Systems," Accepted to IEEE Transactions on Cloud Computing, Jun 2019.
  - Alabbasi, Aggarwal, and Ra, "Multi-tier Caching Analysis in CDN-based Over-the-top Video Streaming Systems," IEEE/ACM Transactions on Networking, vol. 27, no. 2, pp. 835-847, April 2019.

OTHER APPLICATIONS

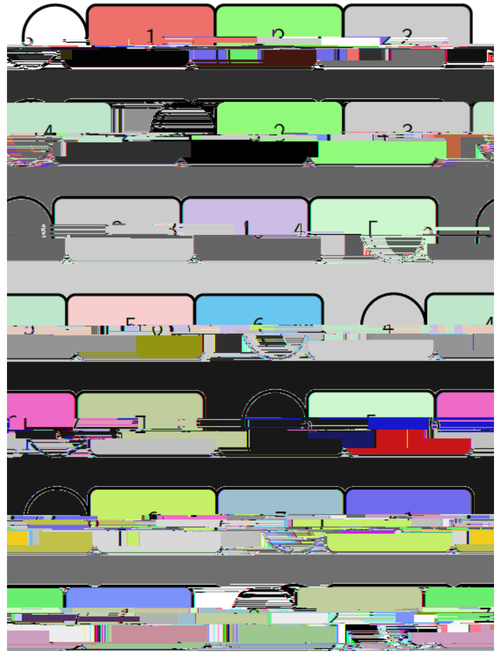
- Caching
- Video streaming over Cloud
- Memory-constrained system
- Coded Computing

MEMORY CONSTRAINED SYSTEM



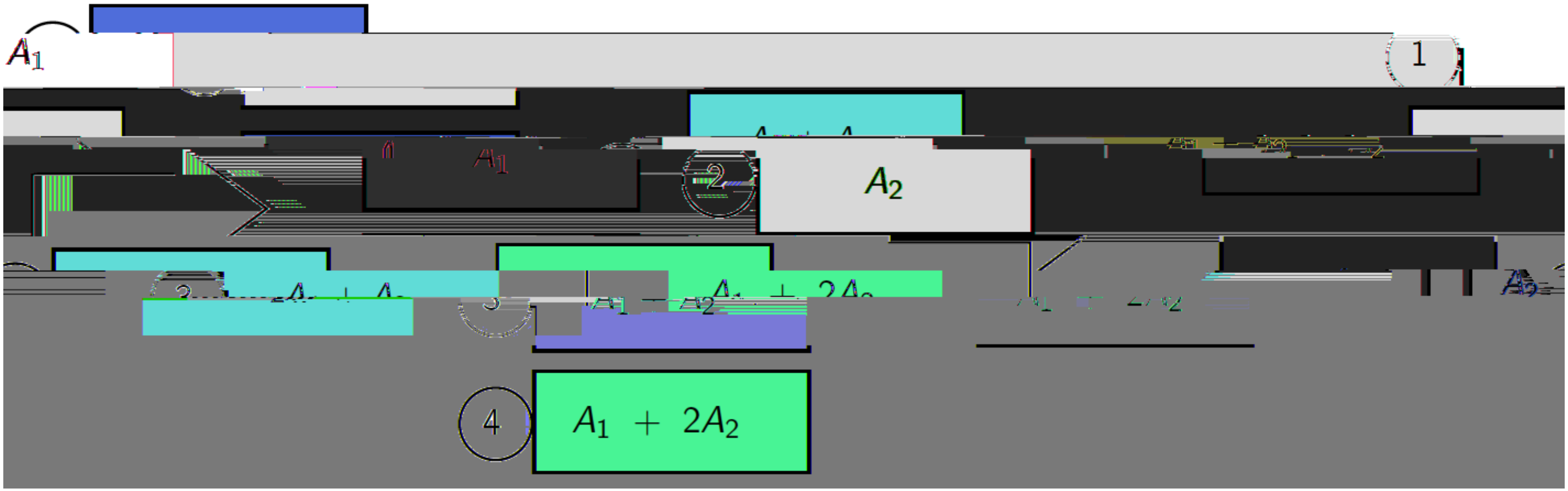
STORAGE MODEL: PLACEMENT

LATENCY OPTIMAL STORAGE AND ACCESS



MDS CODED STORAGE

DECODING COMPLEXITY



SUMMARY

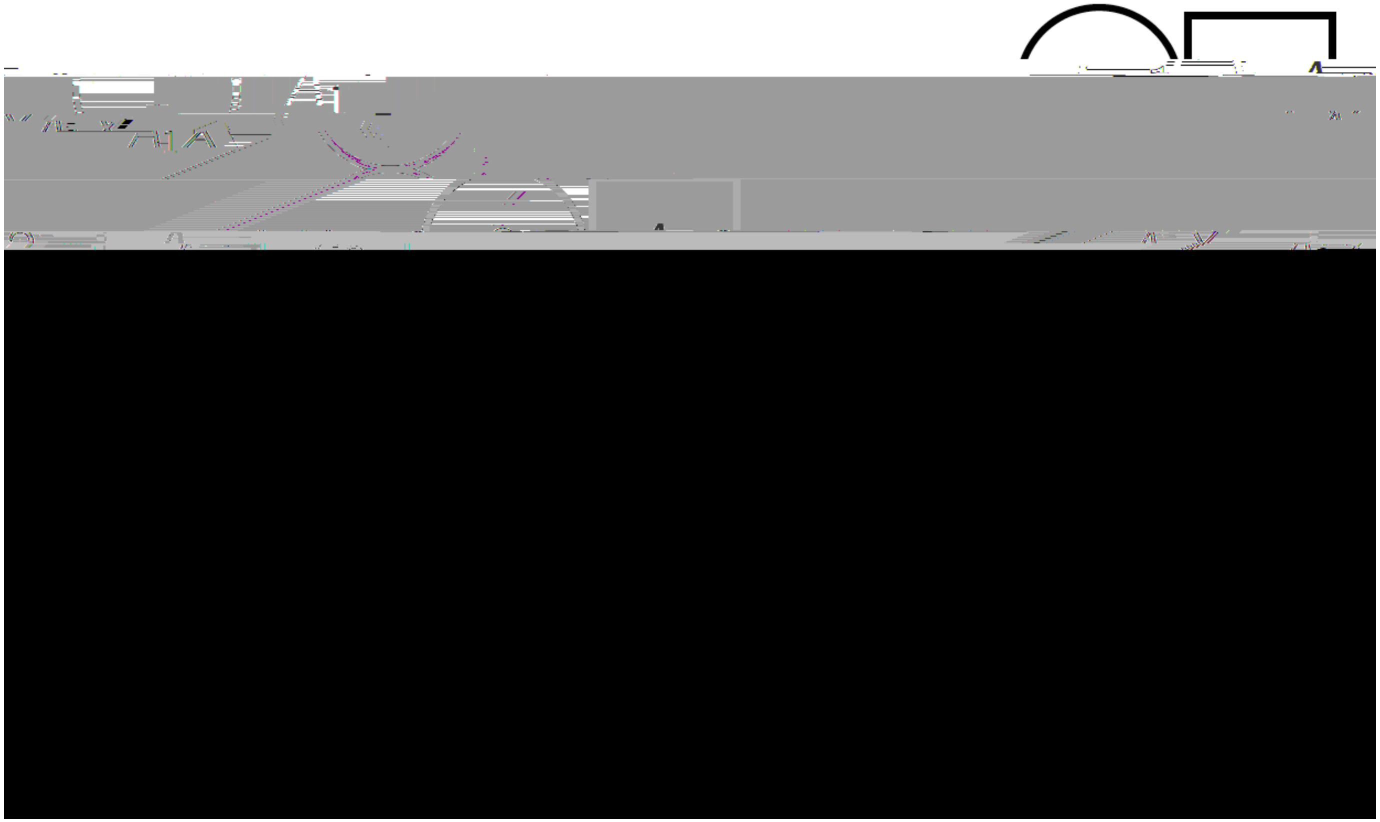
- MDS coded storage is optimal for subfragmented storage
- Subfragmentation of file can lead to competitive performance of replication coded storage
- When storage nodes have no memory constraints all coded storage have identical latency performance
- Staircase coded storage
  - Bitar, Parag, and Rouayheb, ``Minimizing latency for secure coded computing using secret sharing via staircase codes,'' *IEEE Transactions on Communications*. 68(8):4609–4619, Aug 2020.
- Replication coded storage
  - Jinan, Badita, Sarvepalli, Parag, ``Latency optimal storage and scheduling of replicated fragments for memory-constrained servers,'' preprint, 2021.

OTHER APPLICATIONS

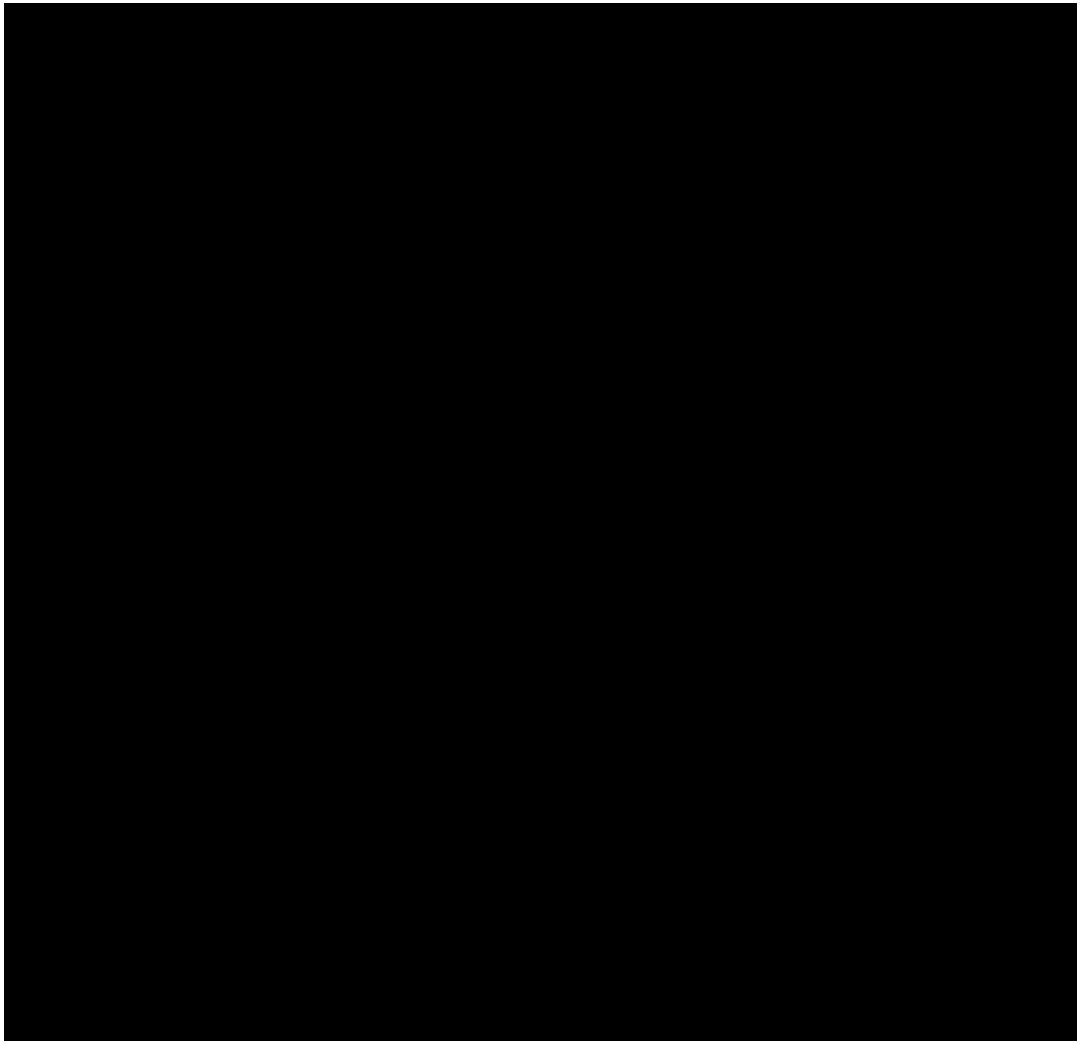
- Caching
- Video streaming over Cloud
- Memory-constrained system
- Coded Computing

MATRIX MULTIPLICATION

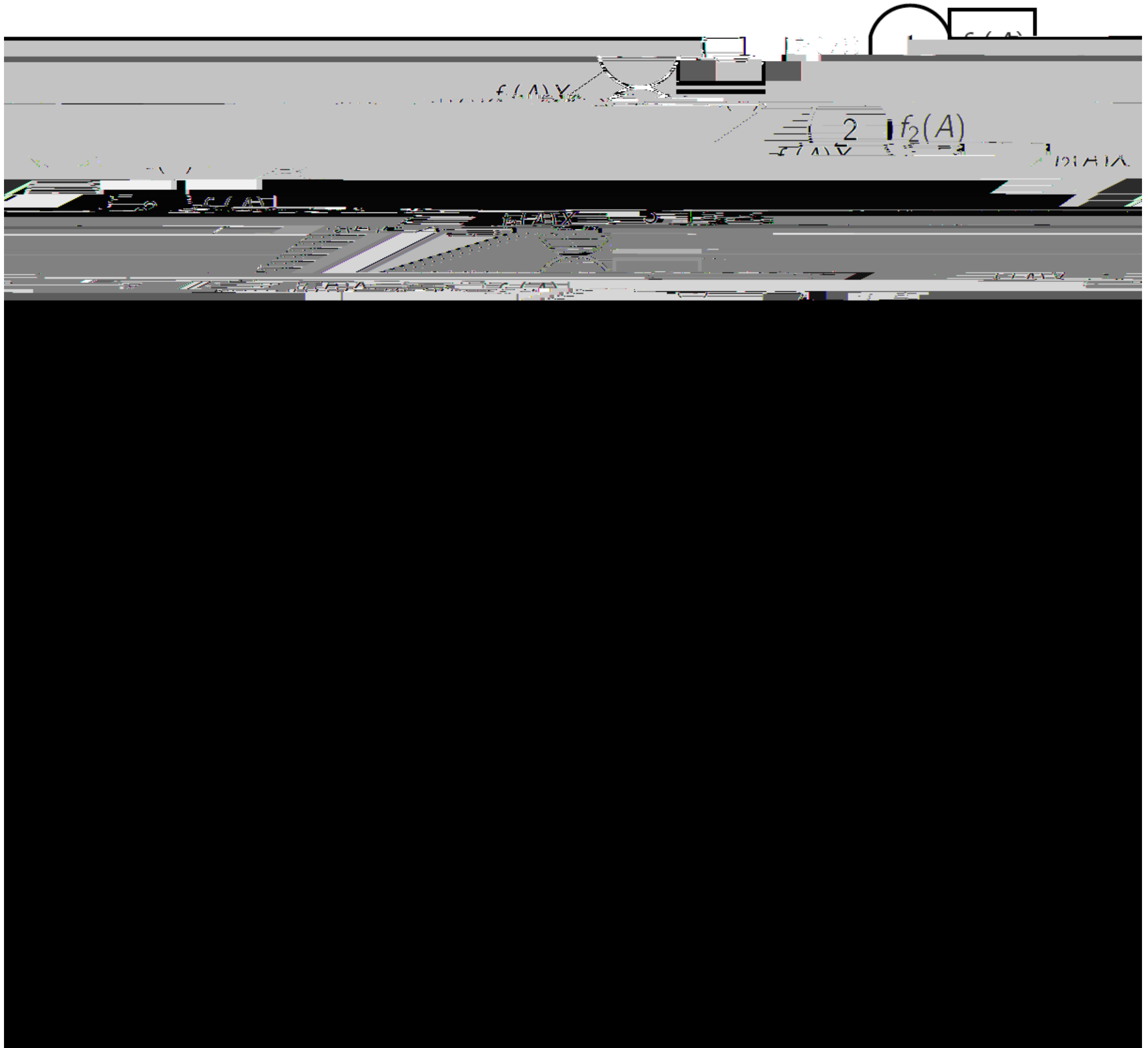
DISTRIBUTED MATRIX MULTIPLICATION



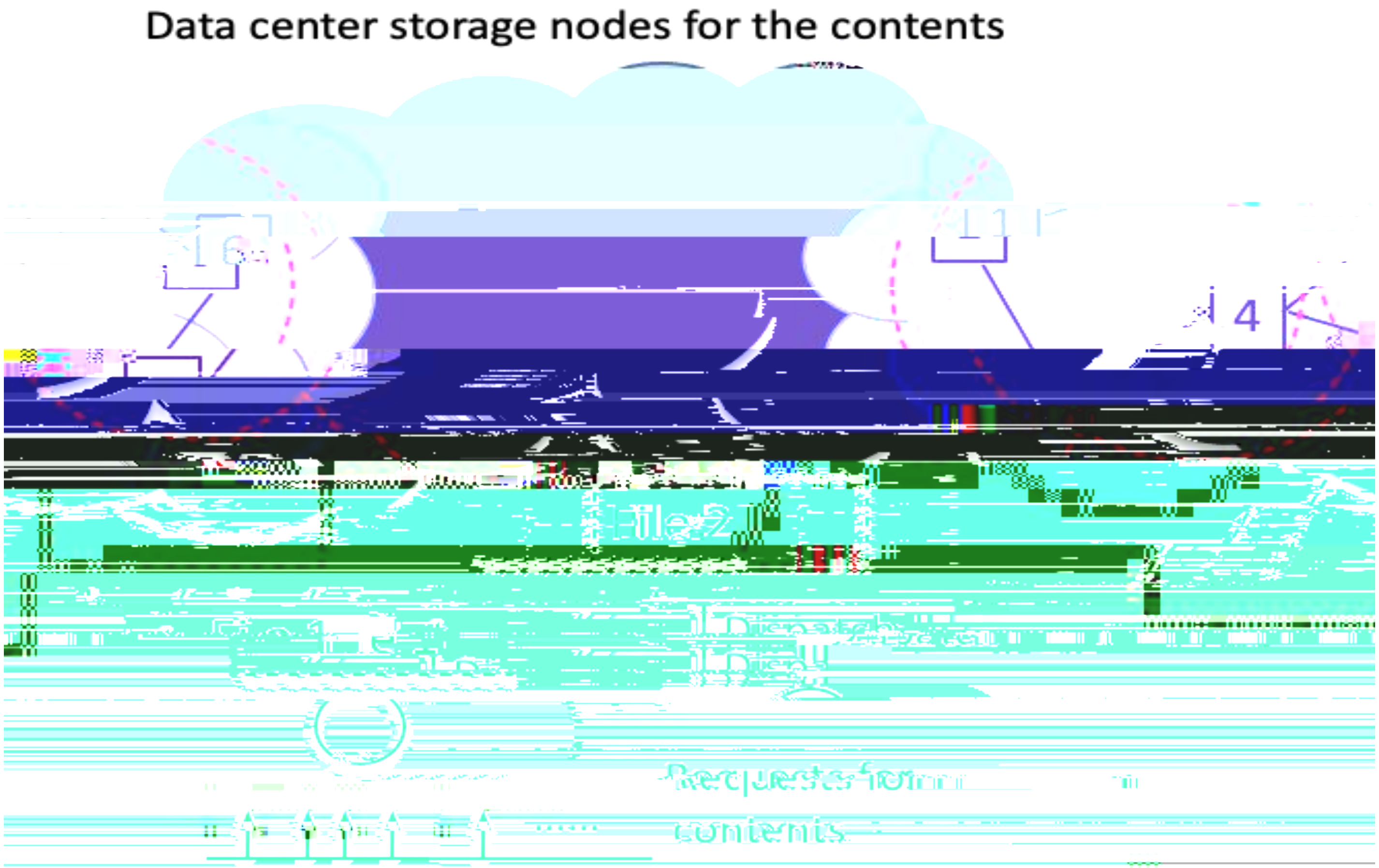
REDUNDANCY FOR STRAGGLER MITIGATION



SUMMARY



KEY PROBLEM IN THIS TUTORIAL



COMPARISON OF STATE-OF-ART: ASSUMPTIONS

[illegible]

COMPARISON OF KEY SCHEDULING STRATEGIES

NUMERICAL COMPARISON OF KEY SCHEDULING STRATEGIES

THANK YOU