C2DN: How to Harness Erasure Codes at the Edge for Efficient Content Delivery

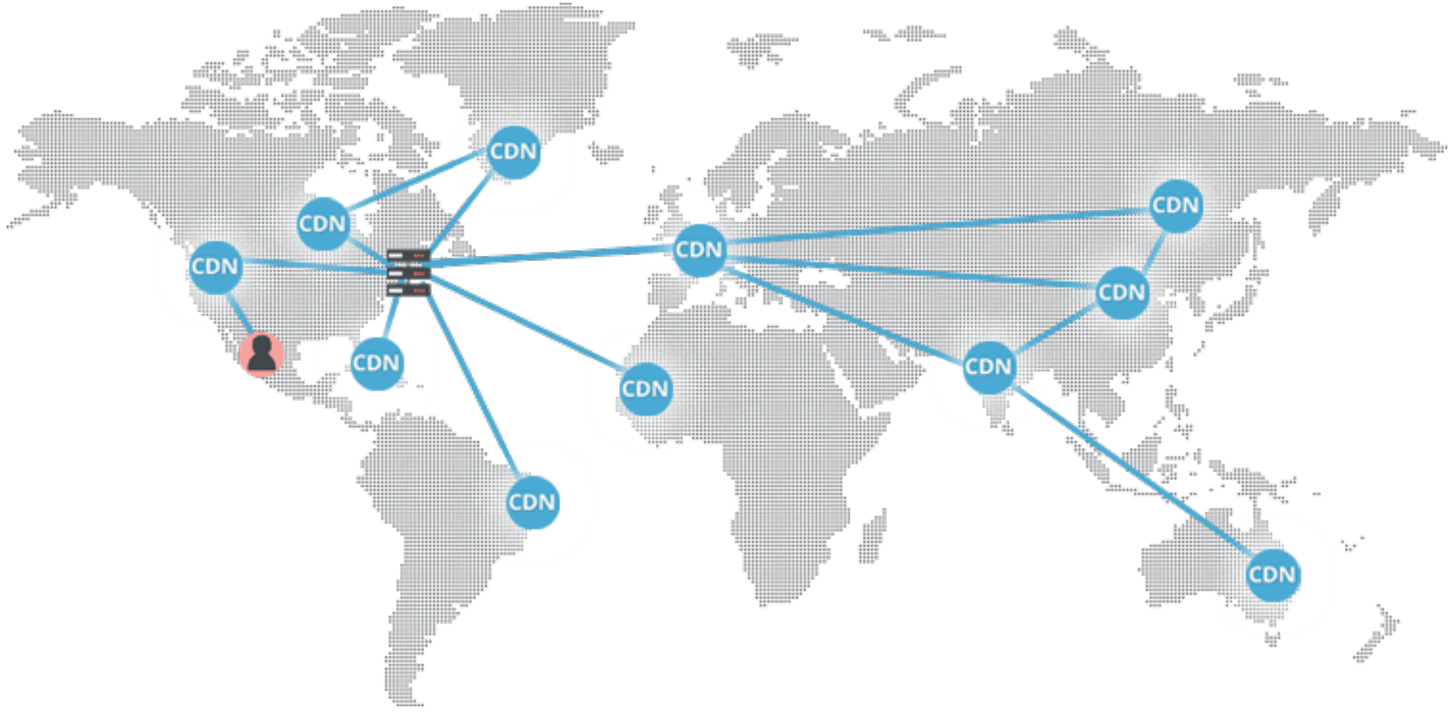
Juncheng Yang*

Anirudh Sabnis#, Daniel S. Berger&, K. V. Rashmi*, Ramesh K. Sitaraman#$

*Carnegie Mellon University, #University of Massachusetts, Amherst, $Akamai, &University of Washington, Microsoft Research

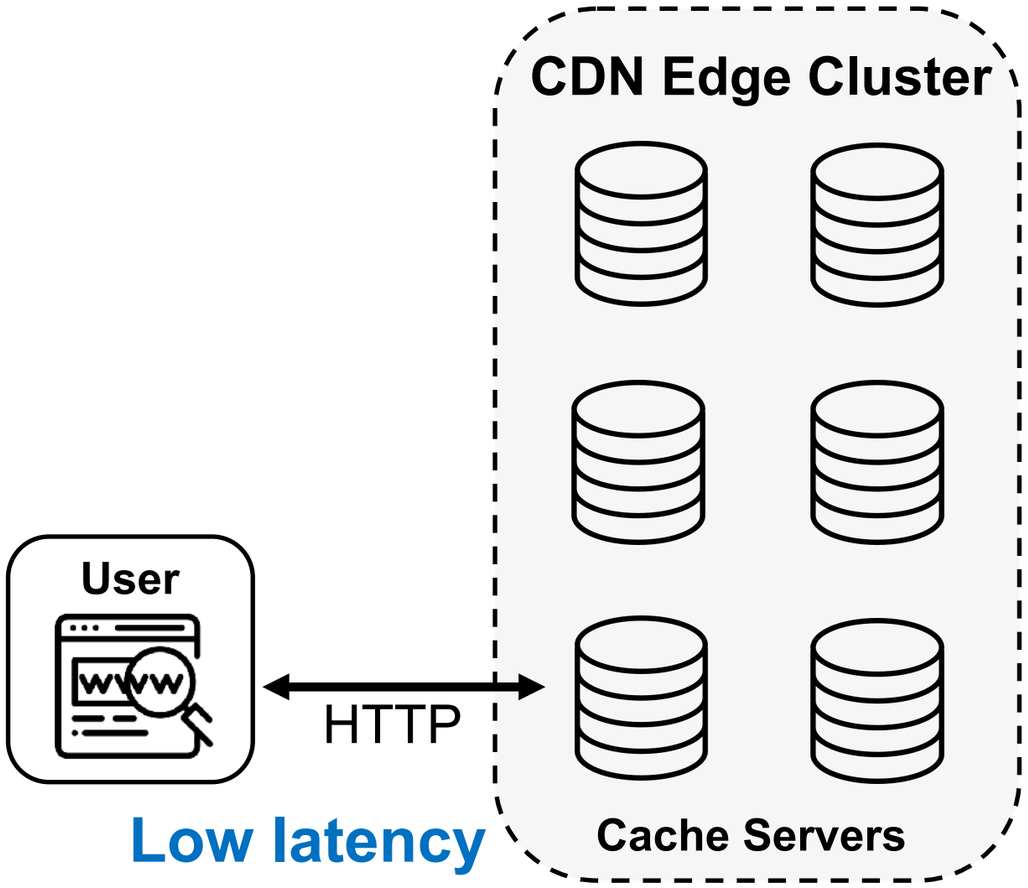
Content Delivery Networks (CDNs)

- Networks of cache clusters close to users
- 72% of Internet traffic



https://elitestrategies-elitestrategies.netdna-ssl.com/wp-content/uploads/2015/08/CDN-on-page-seo.png

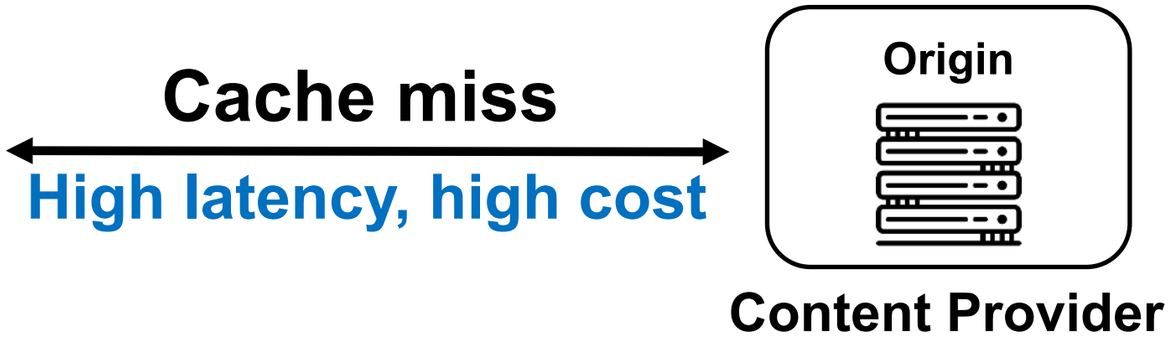
Cache misses are **expensive**



Goal: minimize cache misses (mean and tail)

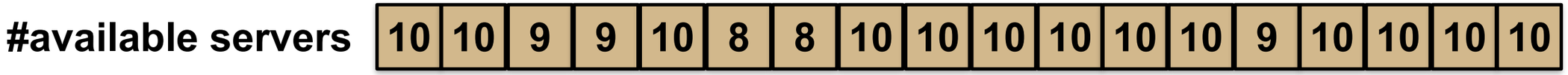
Why?

- Long latency (poor user experience)
- Wide area traffic (cost for CDN and content provider)



Unavailabilities at the edge are common

A month-long trace of 2190 clusters



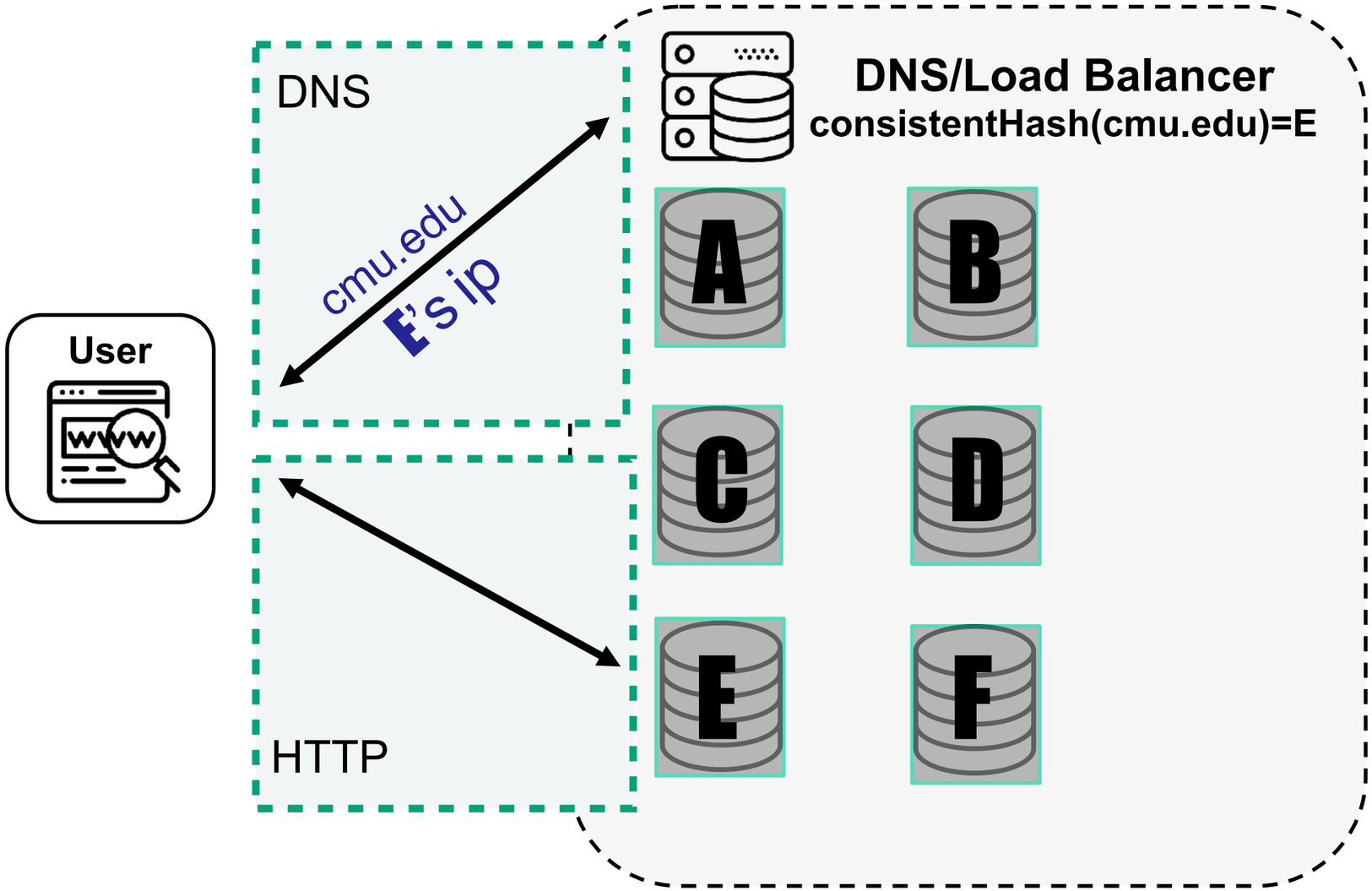
All clusters: **unavailability in 45.2% of observations**

10-server clusters: **unavailability in 30.5% of observations**

Unavailabilities are more common than in datacenters

Reasons: server overload, hardware failure

Bucket-based routing (coarser load balancing)



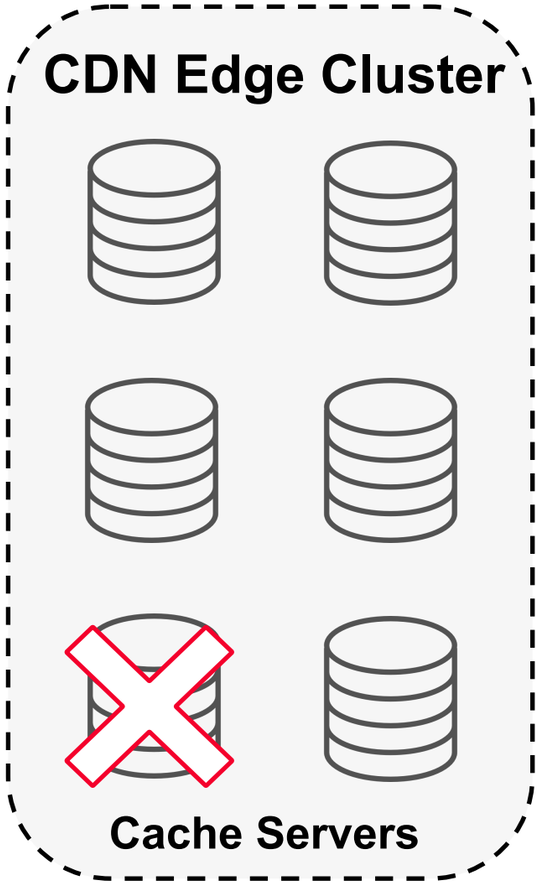
DNS:

cmu.edu	-> E
usenix.org	-> D
cnn.com	-> D

...

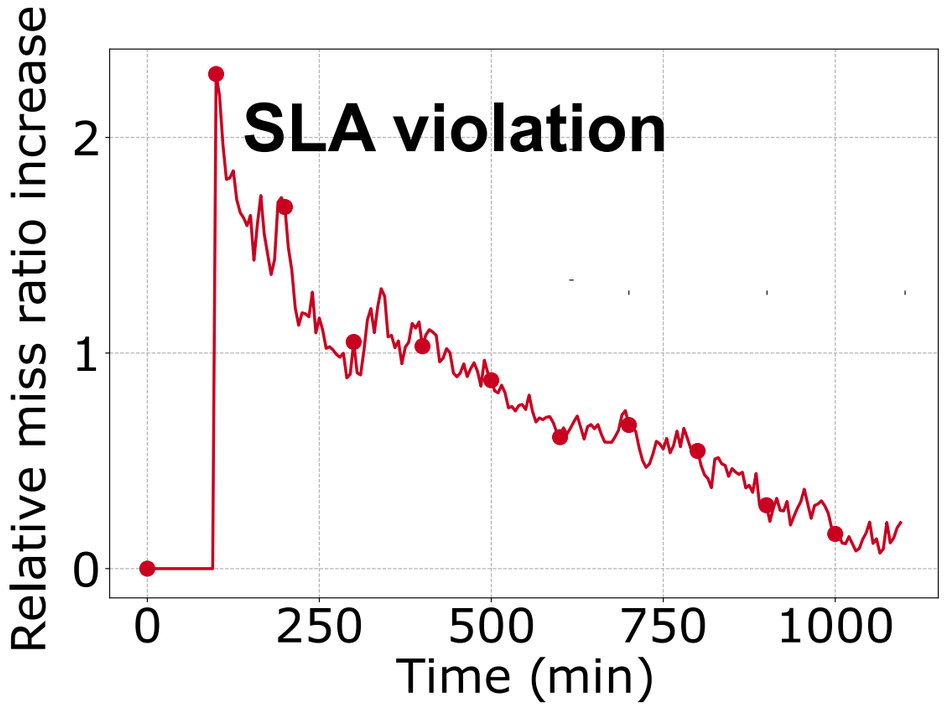
bucket: sub domain name

Bucket-based routing makes unavailability worse



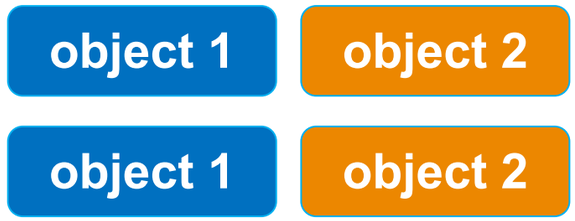
all cached objects for **buckets** mapped to the server become unavailable

miss ratio spike



State-of-the-art solution

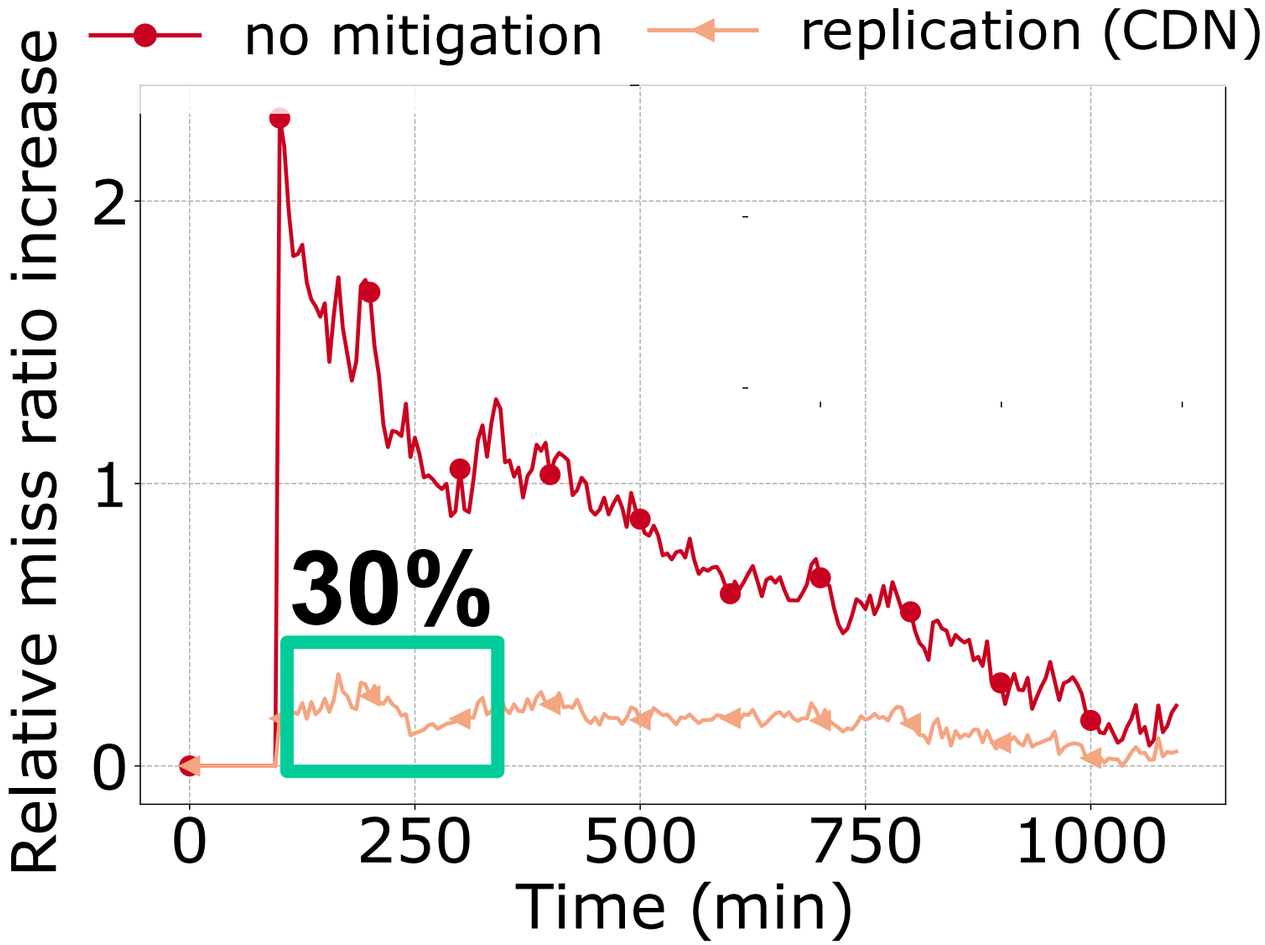
Replication



Limitations

cannot remove spike

Why?

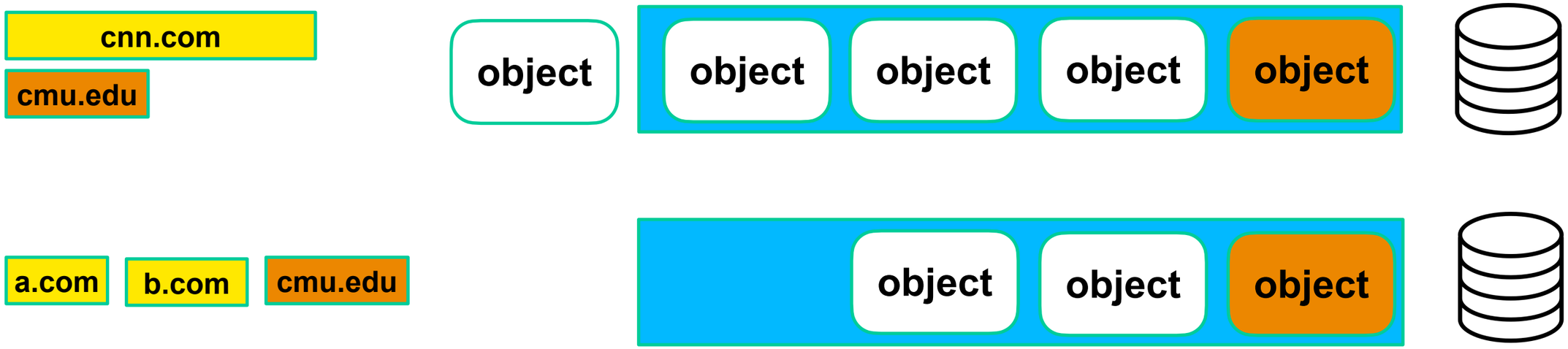


The problem: write load imbalance

Cache writes come from cache misses

Write load is imbalanced, production: max/min server load = 2.5

- Reduces the effectiveness of replication
- SSDs wear out at different rates

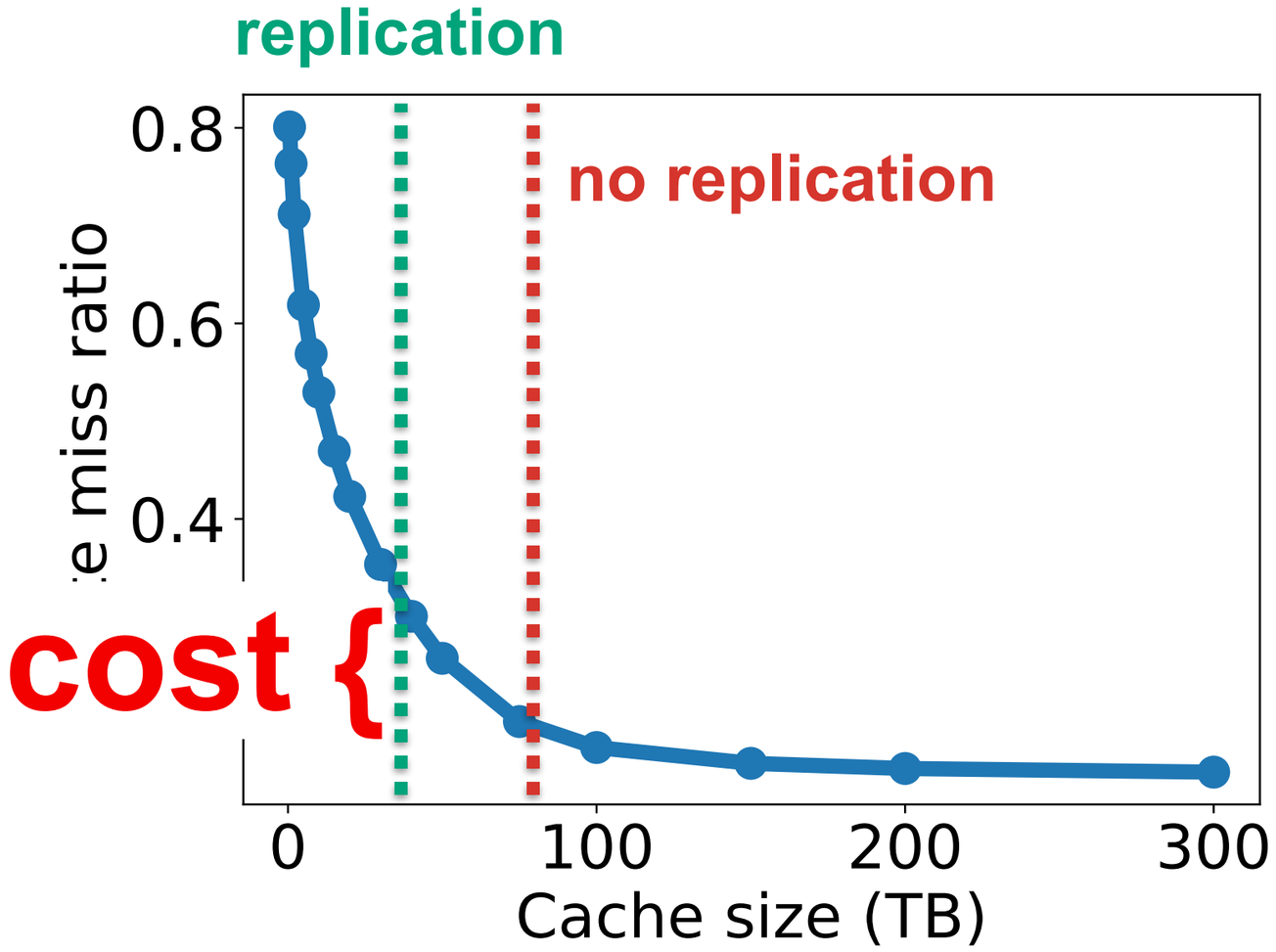


State-of-the-art solution

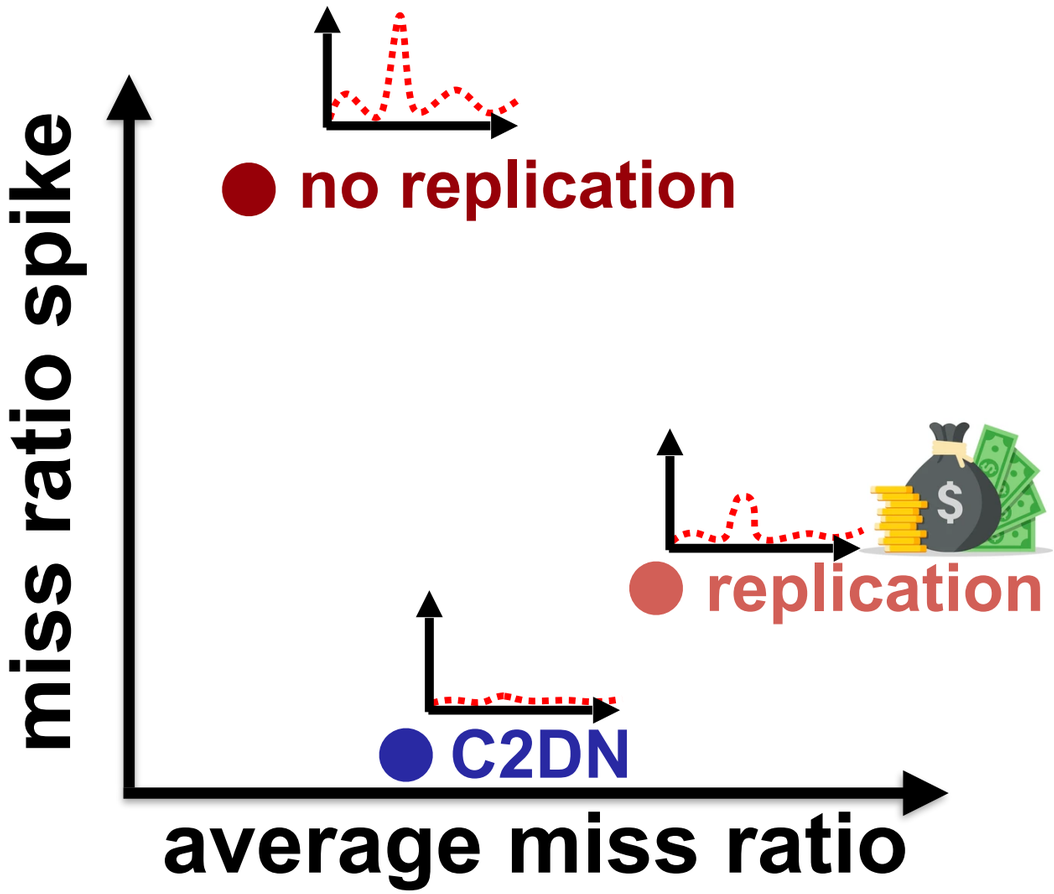
Replication limitations

cannot remove spike

waste limited space



Server unavailability mitigation today is costly and ineffective



- removes miss ratio spikes
- reduces bandwidth cost
- near-perfect write load balancing

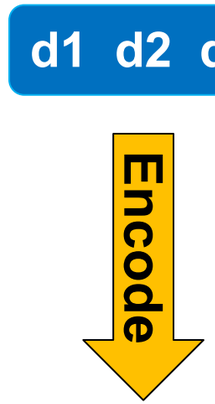
C2DN Design

Erasure coding to reduce storage overhead

Parity rebalance to balance write load

Quick primer on erasure coding: efficient fault tolerance

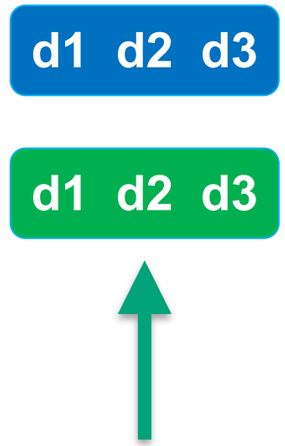
Any ***K*** of ***K+P*** chunks can recover the original data



K data chunks ***P*** parity chunks

33% overhead

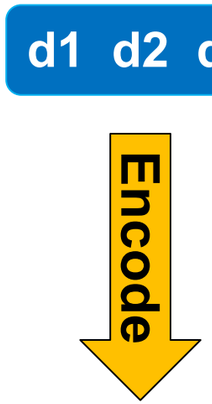
2-way replication



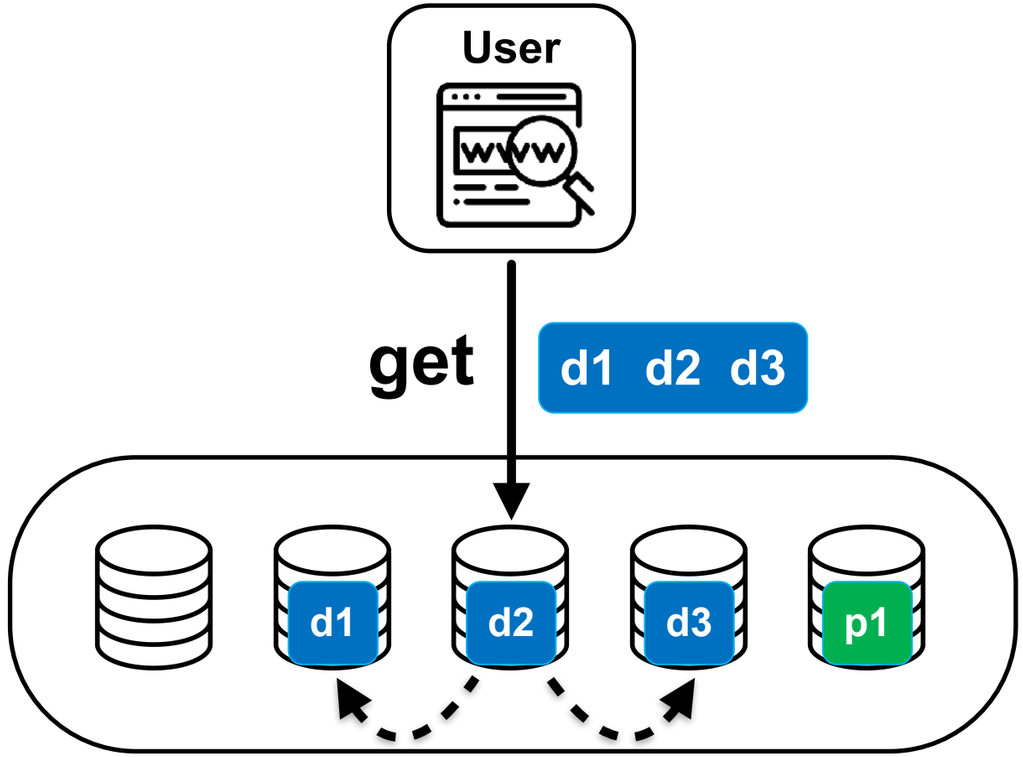
100% overhead

Use erasure coding in CDN clusters?

Any ***K*** of ***K+P*** chunks can recover the original data

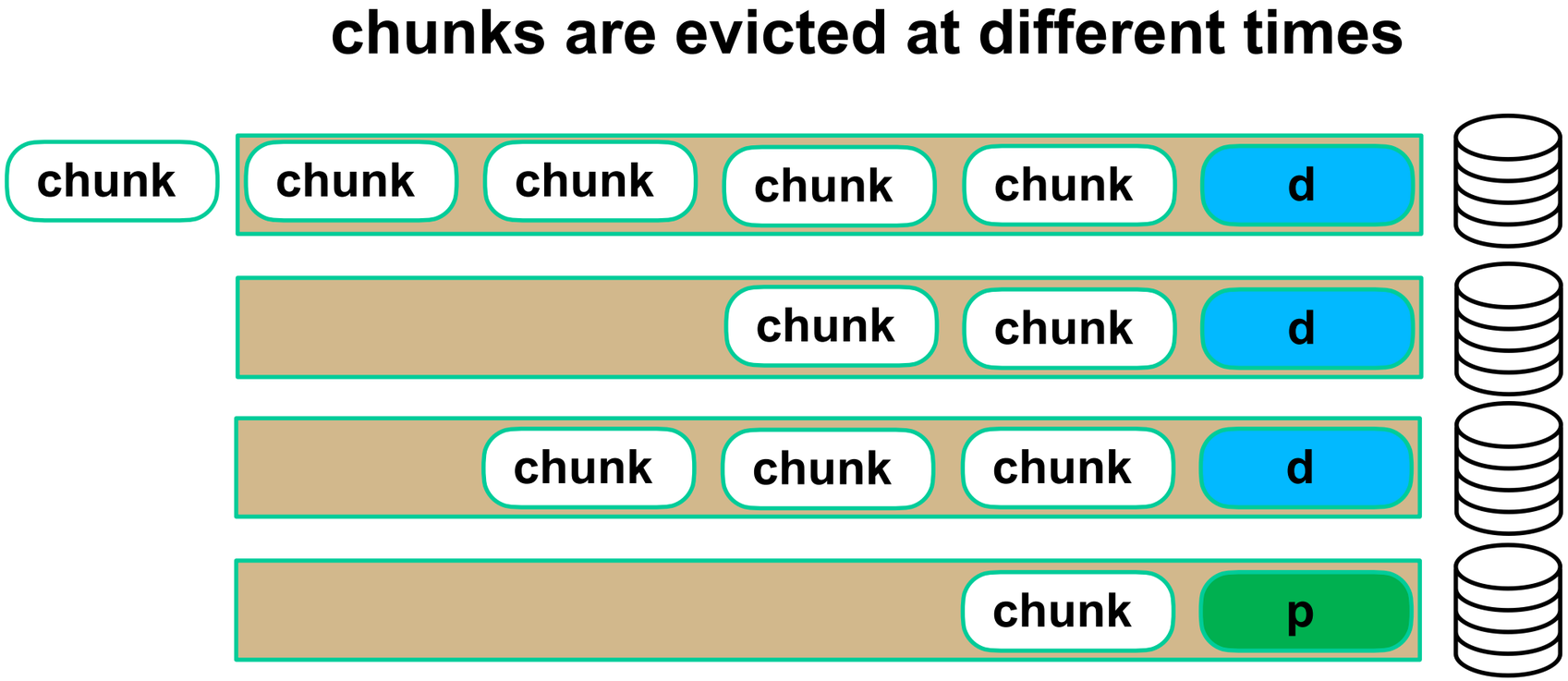


K data chunks ***P*** parity chunks

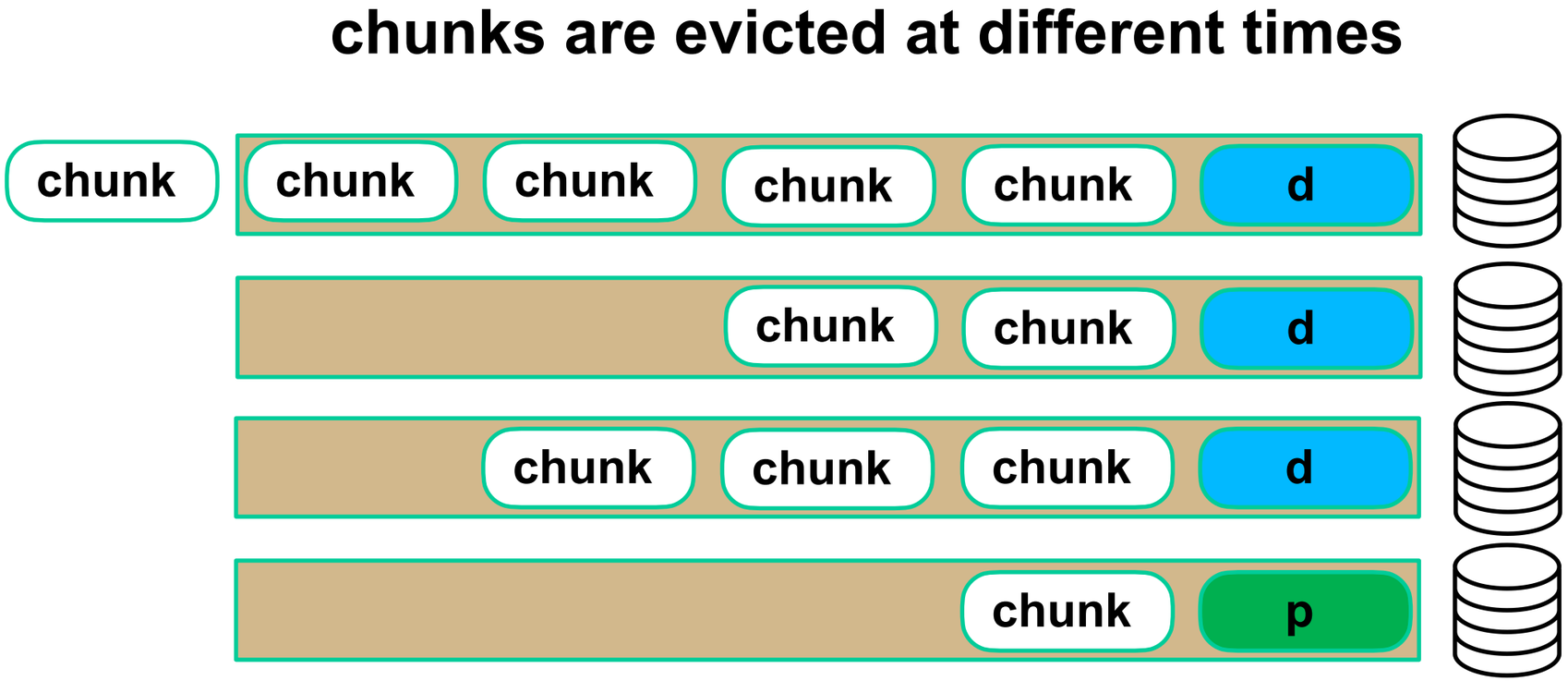


Lower storage overhead, cache more objects!
Lower miss ratio and no unavailability impact?

Naive use of erasure coding is insufficient



Naive use of erasure coding is insufficient

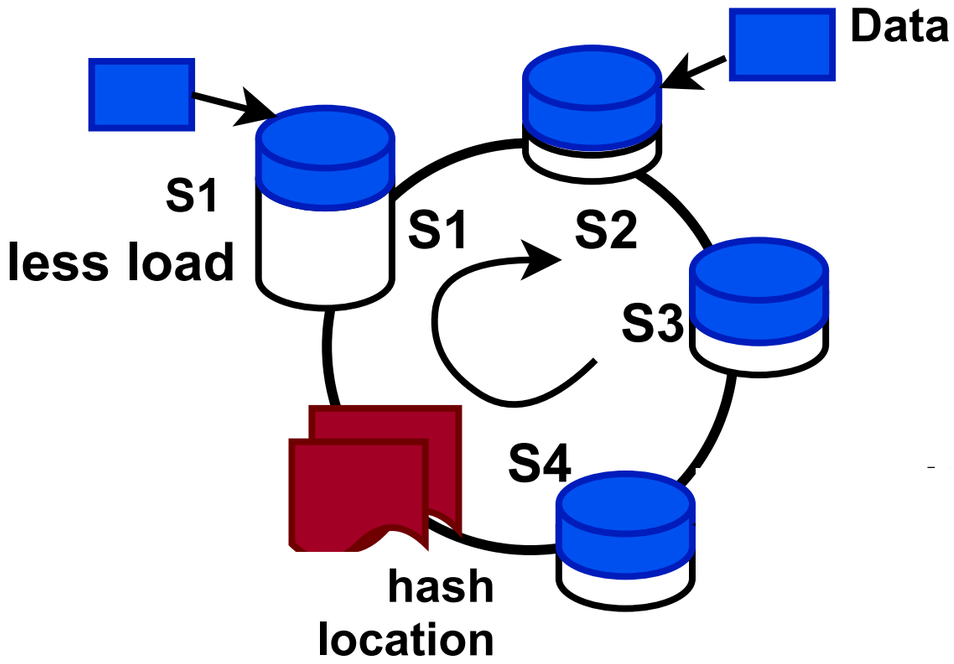


- **Limited miss ratio reduction**
- **Cannot eliminate miss ratio spike** when unavailability happens
- **Runtime overhead** (more CPU usage, longer serving latency)

Goal: rebalance the load

Observation

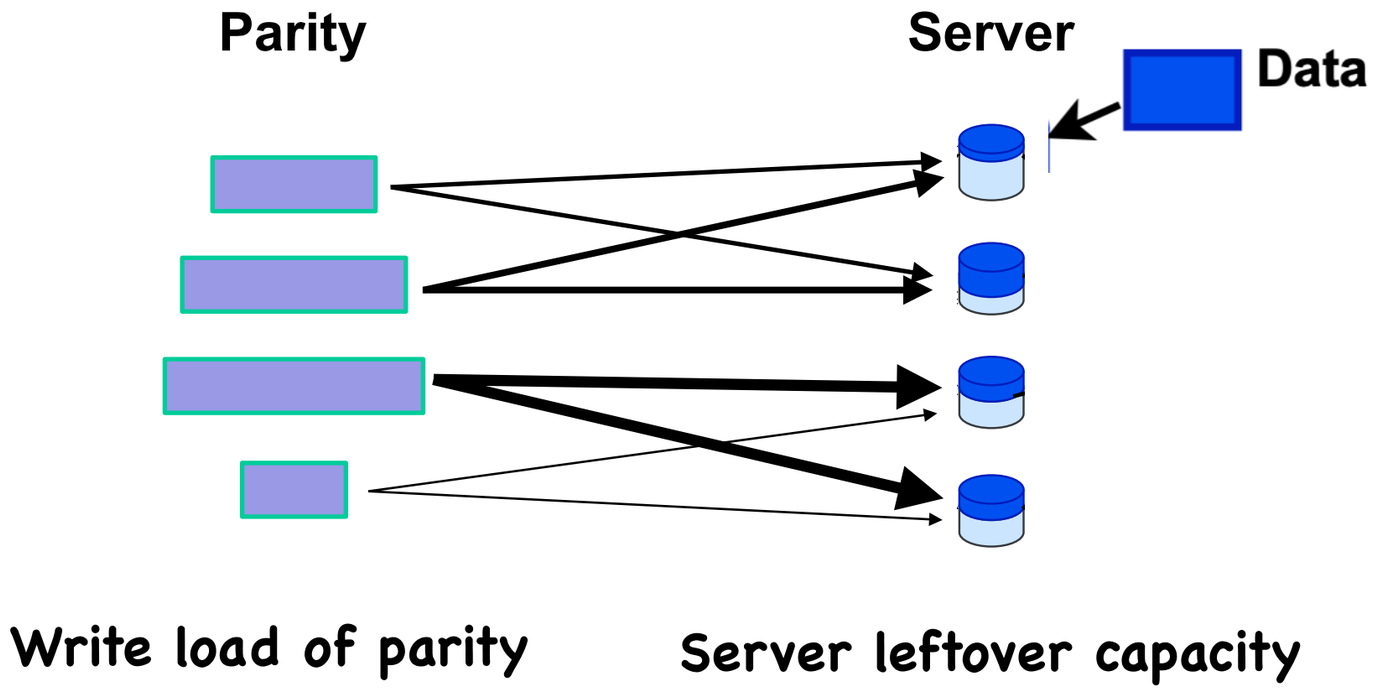
- flexibility in placing parity
- parity chunks are needed **rarely**, **lookup** can be slightly more complex



Technique

- data placement: consistent hashing
- parity placement: rebalance write load

Parity assignment problem



Solution: MaxFlow

Balancing write load: similar cache eviction and SSD wear out rates

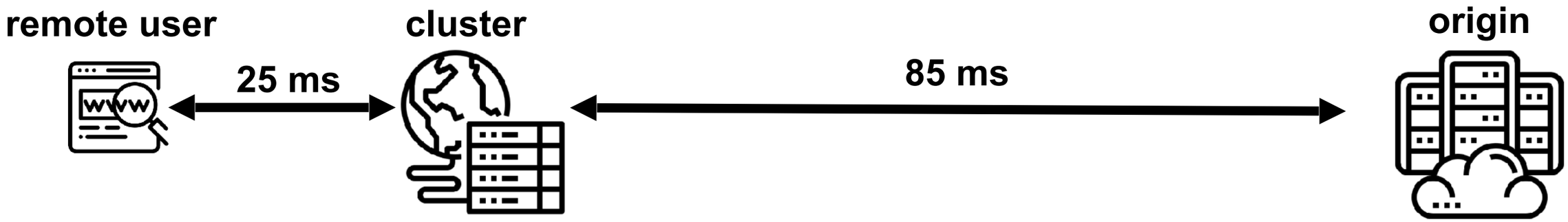
In the paper

- Several other techniques and optimizations
  - hybrid redundancy
  - sub-chunking
  - transparent coding
  - …

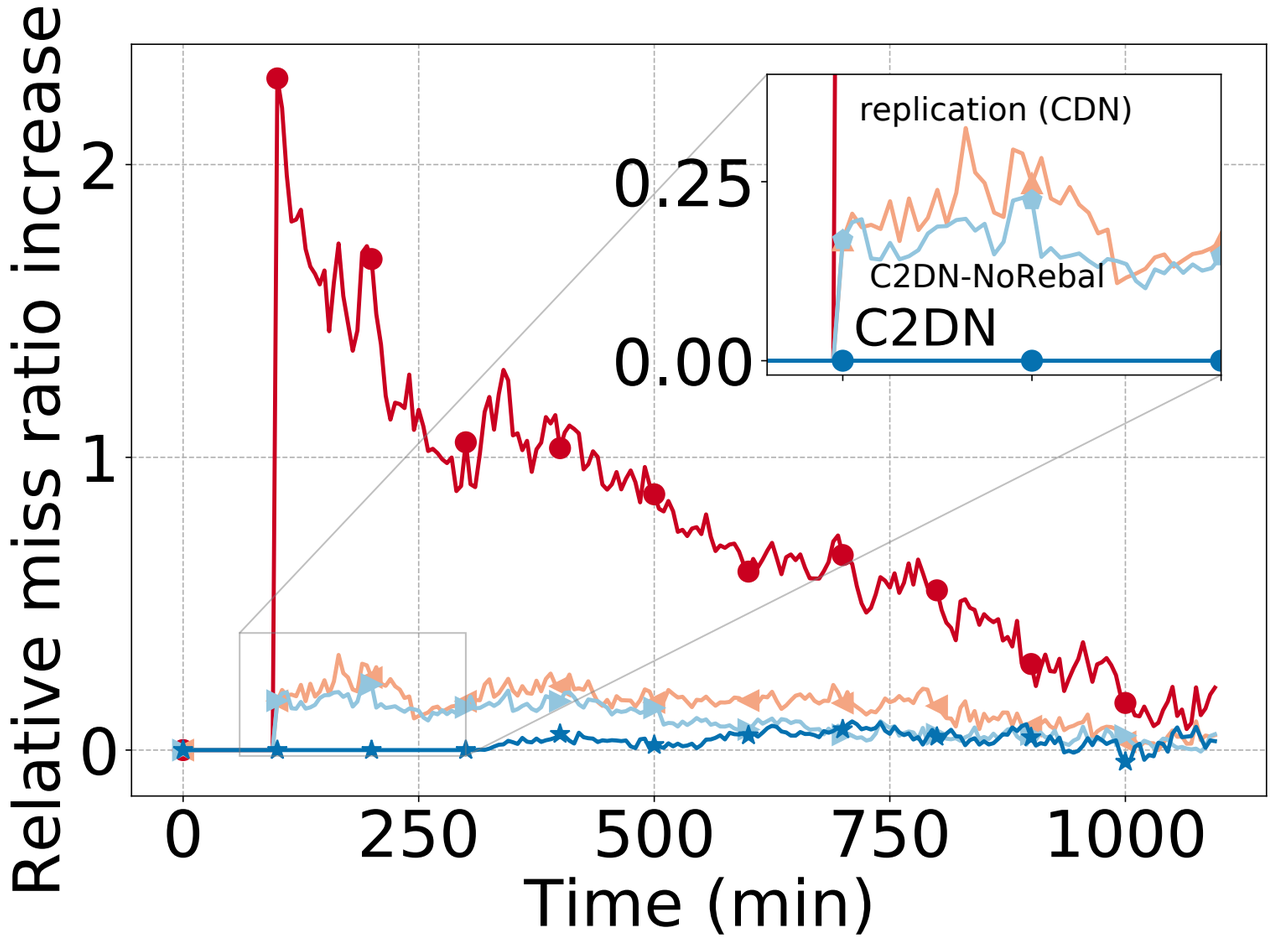
C2DN evaluation

Evaluation setup

- Built C2DN using Apache Trafficserver
- Replayed week-long production traces from Akamai
- Evaluated using three AWS regions



No more miss ratio spike

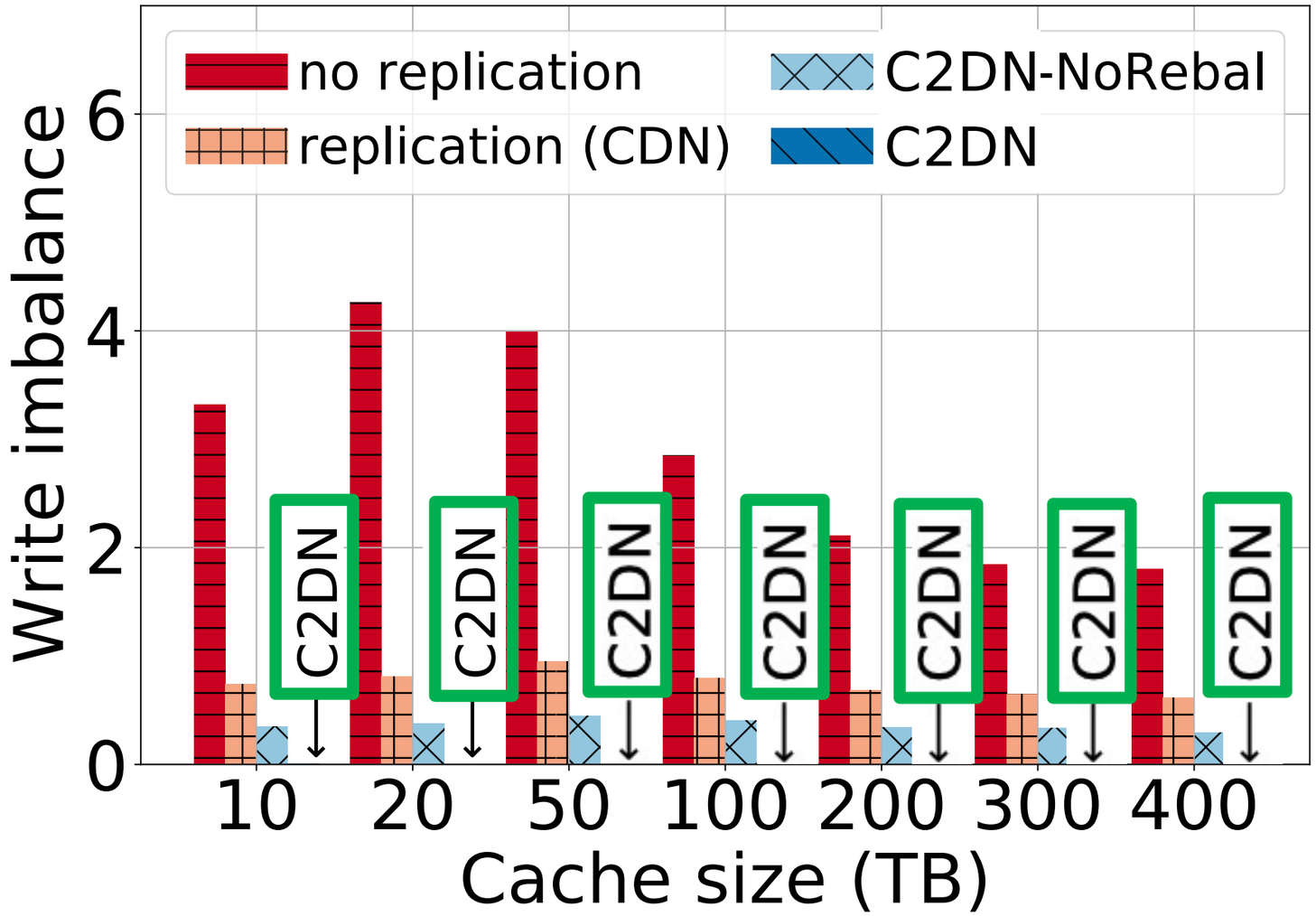


Reducing normal case miss ratio

Cache size	Miss ratio (bandwidth) reduction
production	21%
2x production	16%
4x production	5%

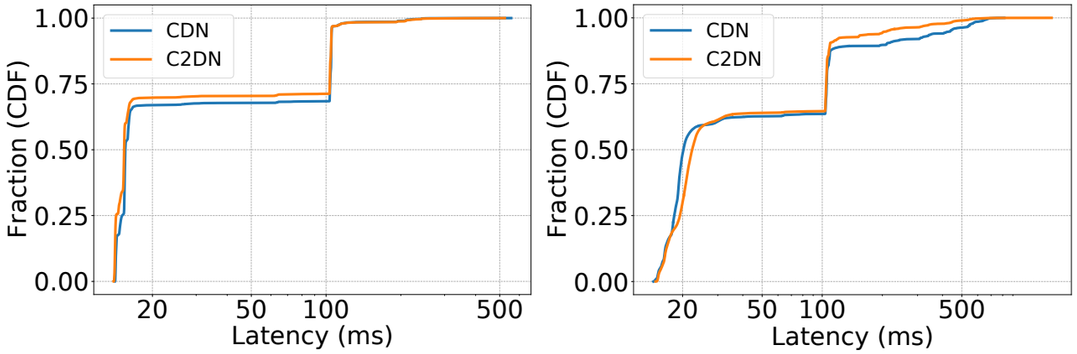
Erasure coding reduces storage overhead
Parity rebalance allows chunks to be evicted at similar time

Near-perfect write load balancing

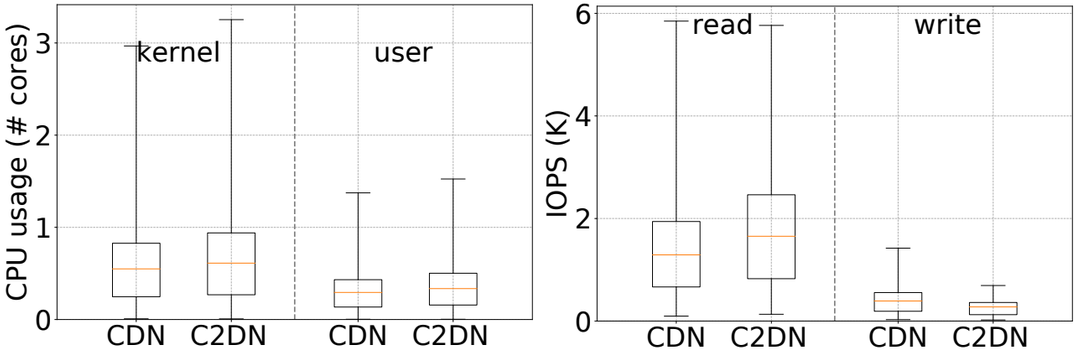


In the paper

- Time-to-first-byte latency and content download time
  - no noticeable latency change



- CPU and disk usage
  - manageable increase

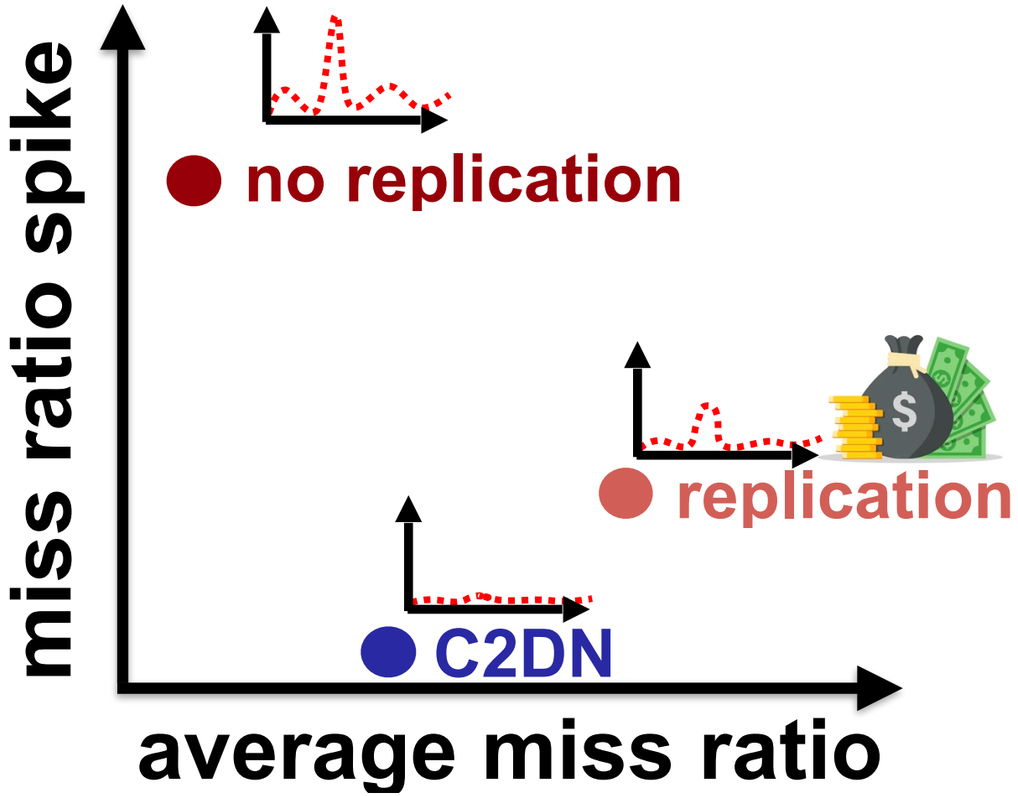


- …

Summary

Unavailability and write load imbalance are common in CDN edge clusters

Traditional approach for fault tolerance is not effective in caching



Open sourced at
https://github.com/Thesys-lab/C2DN

Questions?