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
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

#available servers:

<table>
<thead>
<tr>
<th>Available Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 10 9 9 10 8 8 10 10 10 10 10 9 10 10 10 10</td>
</tr>
</tbody>
</table>

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/Load Balancer
consistentHash(cmu.edu)=E

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

bucket: sub domain name
Bucket-based routing makes unavailability worse

**Cache Servers**

**CDN Edge Cluster**

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

**Miss ratio spike**

**SLA violation**

Relative miss ratio increase

0 0.5 1 1.5 2

0 250 500 750 1000

Time (min)
State-of-the-art solution

Replication

- object 1
- object 2
- object 1
- object 2

Limitations

- cannot remove spike

Why?

![Graph showing relative miss ratio increase over time with and without mitigation. The graph indicates a 30% decrease in miss ratio with replication (CDN) compared to no mitigation. The x-axis represents time in minutes, ranging from 0 to 1000, and the y-axis represents relative miss ratio increase. The graph compares the performance of replication (CDN) and no mitigation, highlighting the effectiveness of replication in mitigating miss ratio spikes.]
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

[bubble chart showing replication limitations and cost]
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

2-way replication

$K$ data chunks  $P$ parity chunks

33% overhead  100% overhead
Use erasure coding in CDN clusters?

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

Lower storage overhead, cache more objects! Lower miss ratio and no unavailability impact?
Naive use of erasure coding is insufficient

chunks are evicted at different times
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

<table>
<thead>
<tr>
<th>Cache size</th>
<th>Miss ratio (bandwidth) reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>production</td>
<td>21%</td>
</tr>
<tr>
<td>2x production</td>
<td>16%</td>
</tr>
<tr>
<td>4x production</td>
<td>5%</td>
</tr>
</tbody>
</table>

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
  - ...

---

**Graphs:**

- CDF of latency for CDN and C2DN
- Box plots of CPU usage (kernel and user) and IOPS (read and write) for CDN and C2DN
Summary

Unavailability and write load imbalance are common in CDN edge clusters.

Traditional approach for fault tolerance is not effective in caching.

Questions?

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