CDN Caching

COS 316: Principles of Computer System Design

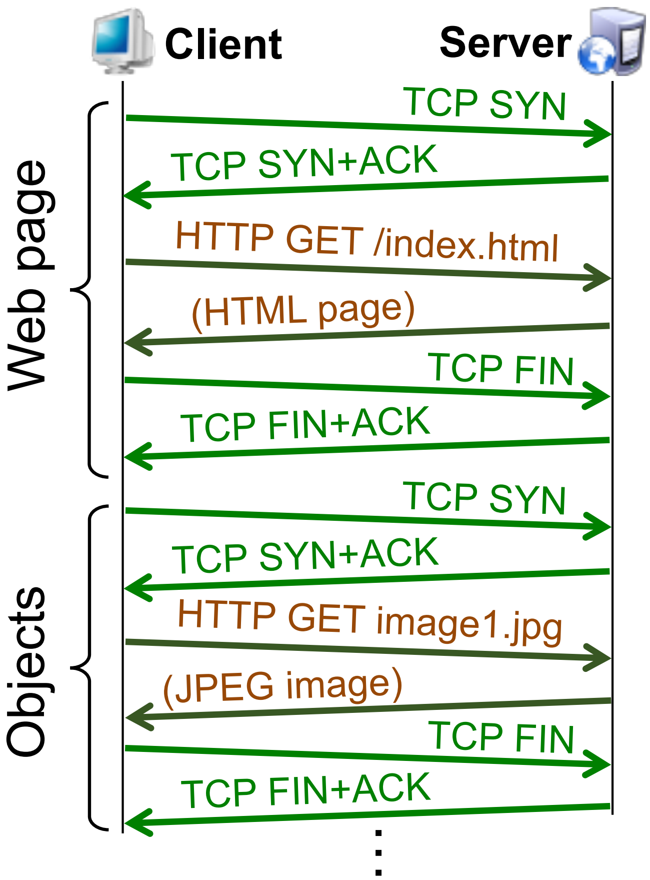
Lecture 9

Amit Levy & Wyatt Lloyd

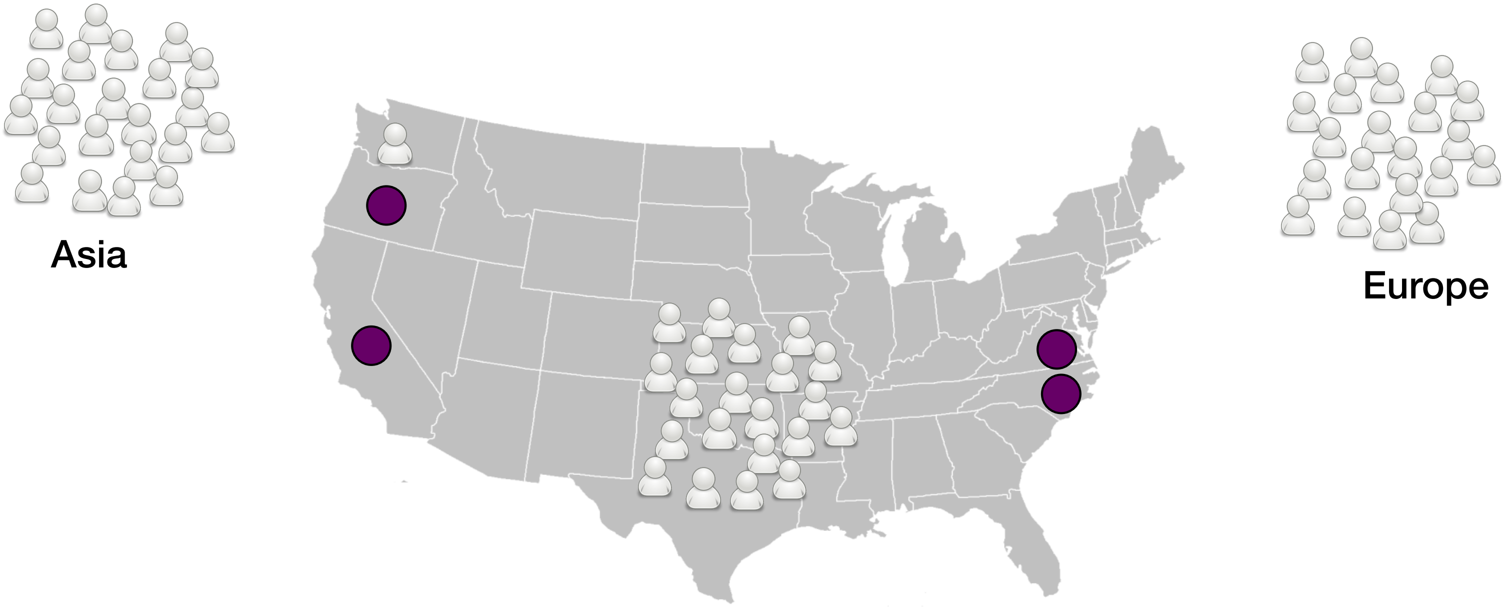
Anatomy of a Web Page Fetch

- Web page = HTML file + embedded images/objects

- HTML page does not embed objects in it
  - *Q: Why not?*

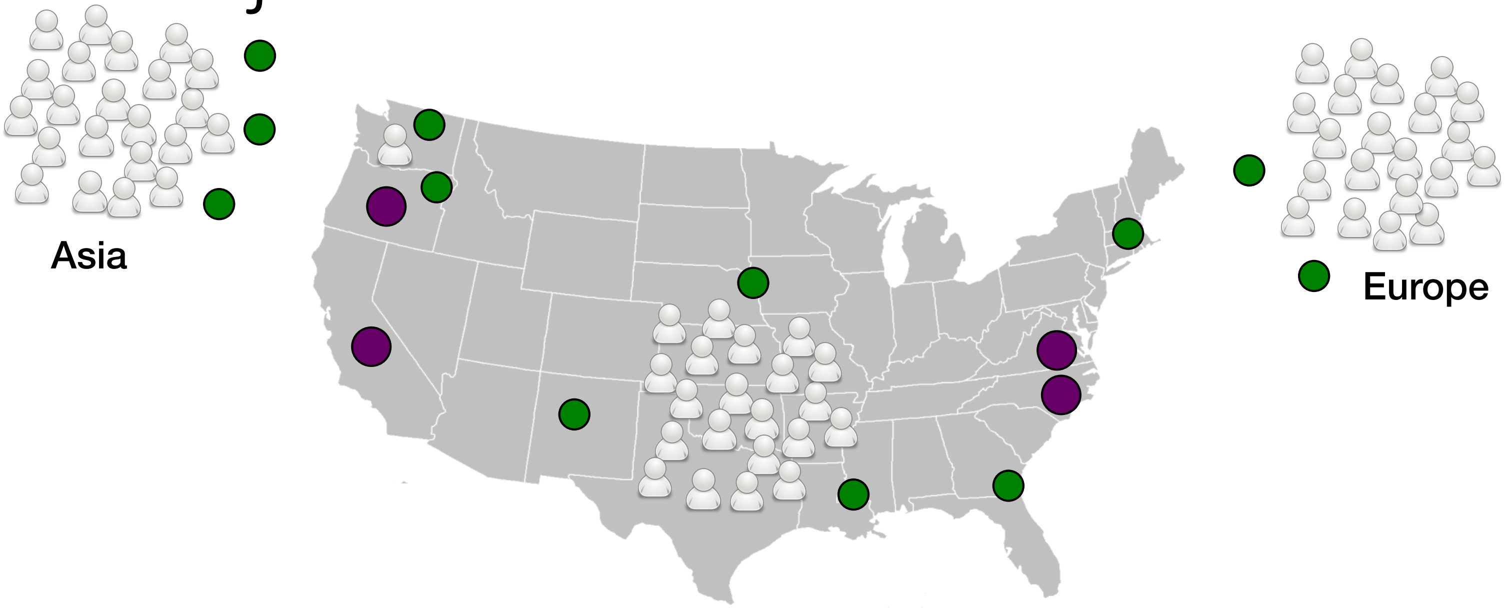


Serve It All From Datacenters?



- High latency for many clients ☹
- Use a ton of bandwidth to send the same objects over and over ☹

Serve Objects From CDN



- Lower latency because content is closer ☺

- Less global bandwidth ☺

CDN Locations

- Where can we stick machines?
- Inside our datacenters
- At colocation facilities
  (other people's DCs)
- At Internet Exchange Points
- Inside others' networks

Not *that* many machines (limited space and/or high cost)

CDN Locations vs Datacenters

- There are many more CDN locations
  - e.g., Akamai has 1000+
- CDNs are closer to people
  - Because there are more of them
  - Placed for locality
- CDN locations are much wimpier
  - Maybe 1 rack (~40 machines)
  - …
  - Maybe 8 racks (~320 machines)

Which ones do clients connect to?

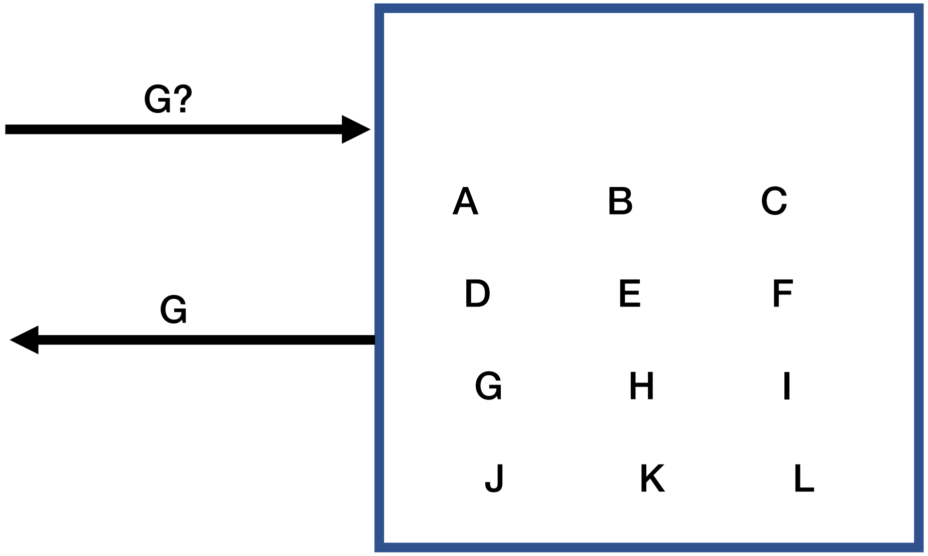
What do we store on them?

CDN Locations Store What?

- Store everything?
  - Exabytes of data for some services
  - 1 rack (40 machines):
    - 144 GB memory? * 40 -> 5.8TB memory … not even close…
    - 10x4 TB HDD? * 40 -> 1.6 PB … not even close…

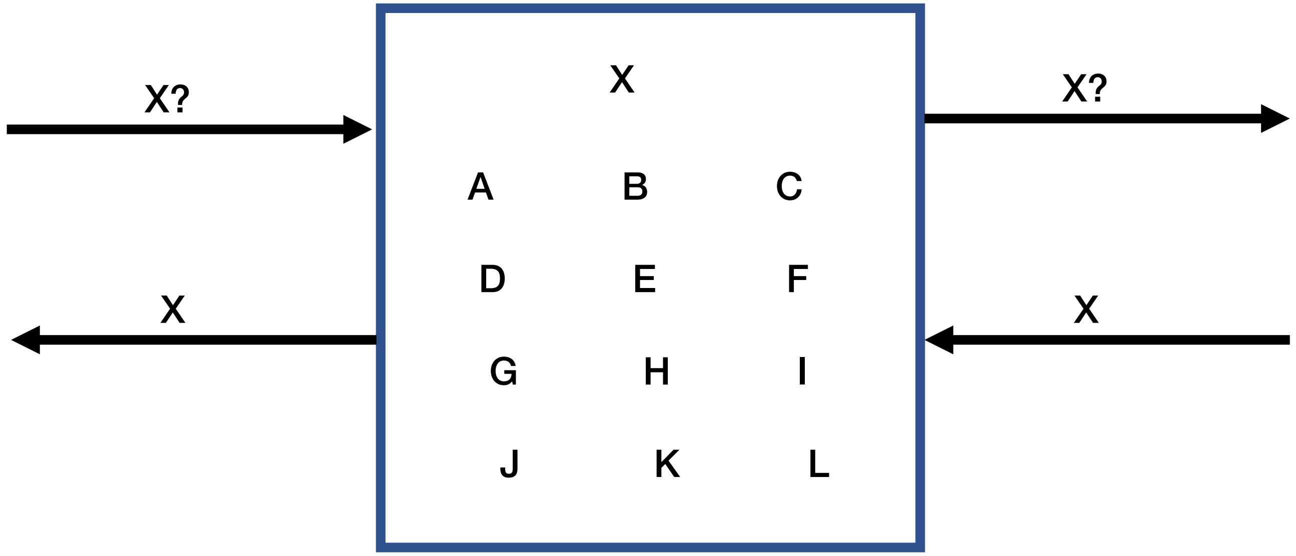
- Need to store a subset of objects!
  - *Q: But which objects to store?*

CDN Cache Hit



A B C
D E F
G H I
J K L

CDN Cache Miss



If I want to store X, what do I get rid of to make space?

Cache Algorithms 101

- First In First Out (FIFO)
- Least recently used (LRU)
- Least frequently used (LFU)
- Belady (Offline optimal)

- (Note: all fully associative today)

First-In-First-Out (FIFO)

- Evict objects added to cache longest ago
- Very simple!

- 3 item cache example:
  - Request stream: a, b, a, c, a, d, a, e, a, f, g

- Can we do better?

Least Recently Used (LRU)

- Evict object used longest ago
  - “Objects used more recently are more likely to be accessed again”
  - Exploits temporal locality

- Implementation: Update access time for every hit

- 3 item cache example:
  - Request stream: a, b, a, c, a, d, a, e, a, f, g
  - Request stream: h, h, h, i, j, k, h

Least Frequently Used (LFU)

- Evict object with fewest hits
  - "Objects used more often are more likely to be accessed again"
  - If tie, use LRU

- Implementation: Update access count for every hit

- 3 item cache example:
  - Request stream: a, b, a, c, a, d, a, e, a, f, g
  - Request stream: h, h, h, i, j, k, h
  - Request stream: l, l, m, n, o, m

Belady (Offline Optimal Caching)

- What is the best a caching algorithm could do?
- Offline: uses knowledge of the future
    - (Can't use in practice)

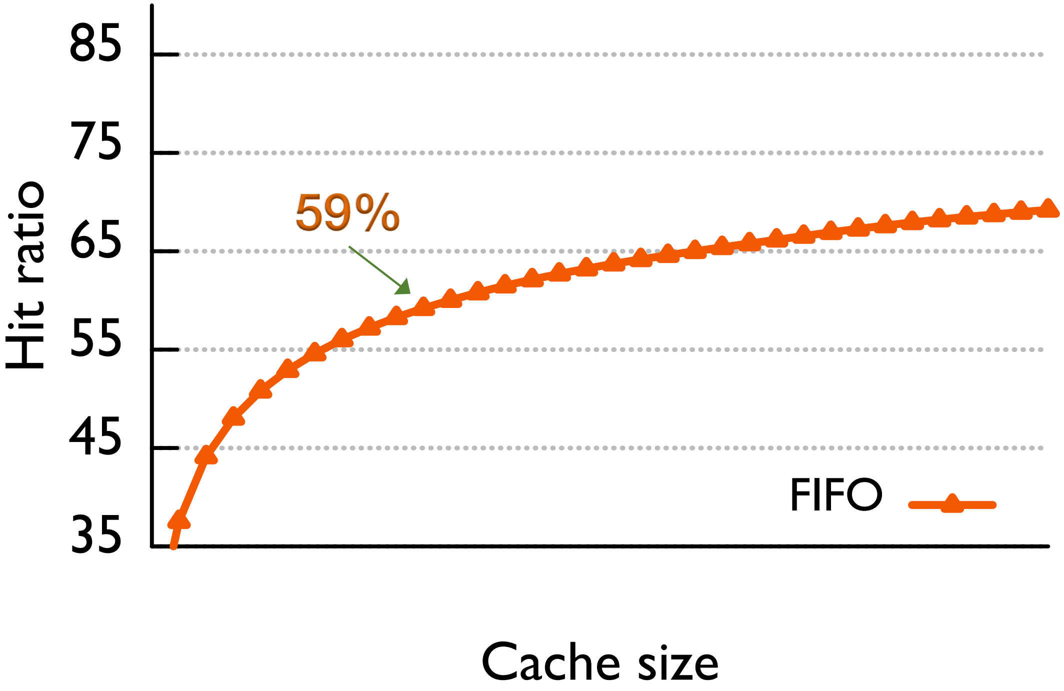
- Evict the object with the furthest **next** access time
    - Worst object to keep in the cache

- 3 item cache example:
    - Request stream: h, h, h, i, j, k, h
    - Request stream: l, l, m, n, o, m

Effectiveness of Algorithms for CDN Caching?

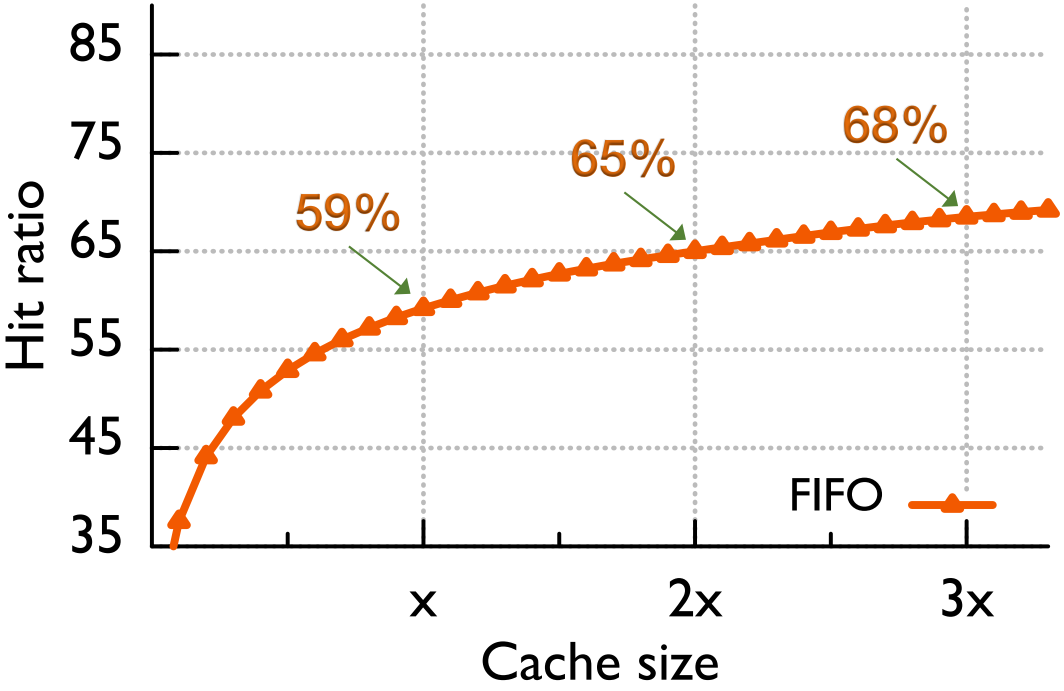
[Figures from Qi Huang’s 2013 SOSP Talk]

Edge Cache with Different Sizes



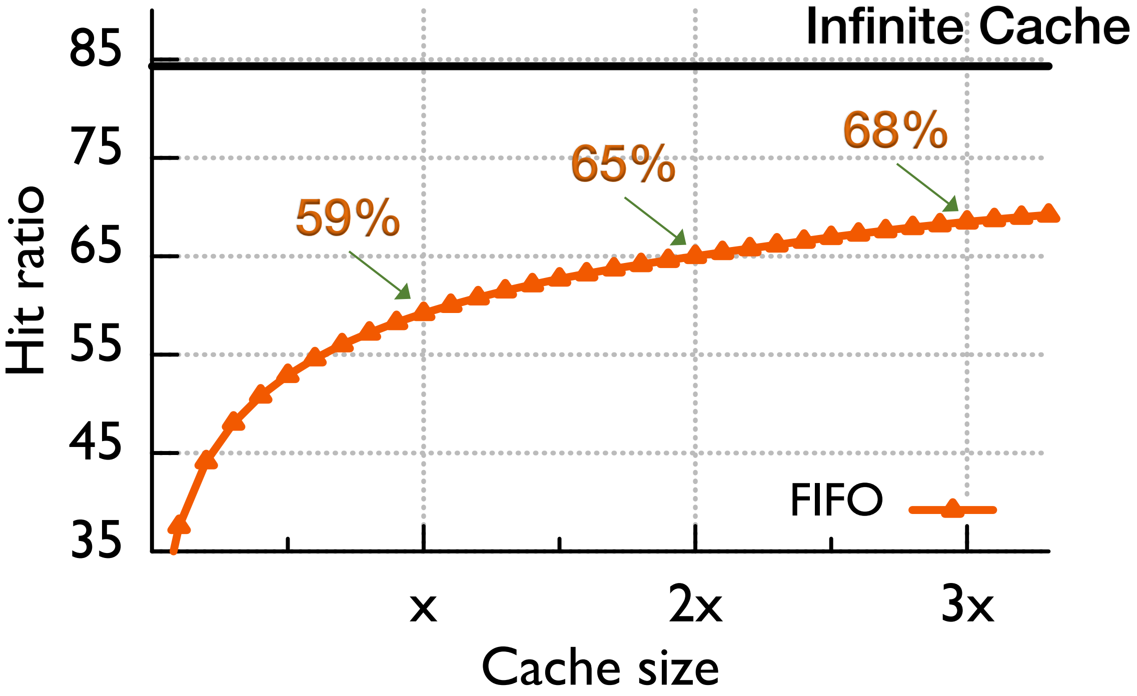
- **Facebook’s San Jose CDN edge cache circa 2013**

Edge Cache with Different Sizes



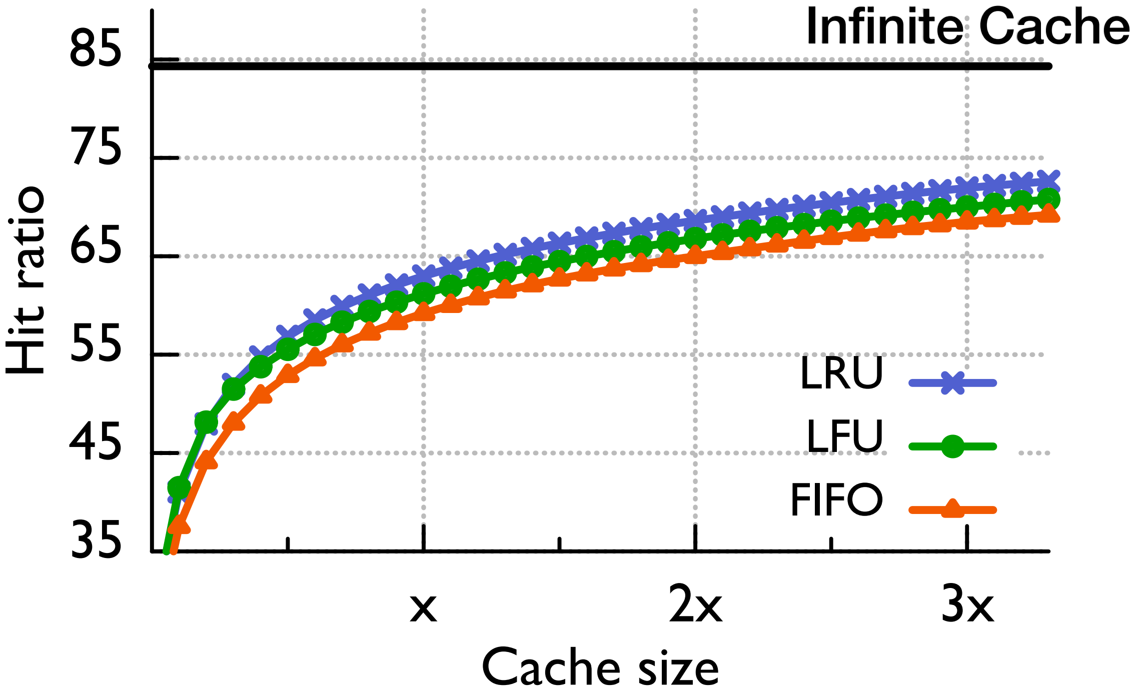
- “x” estimates deployment size (59% hit ratio)

Edge Cache with Different Sizes



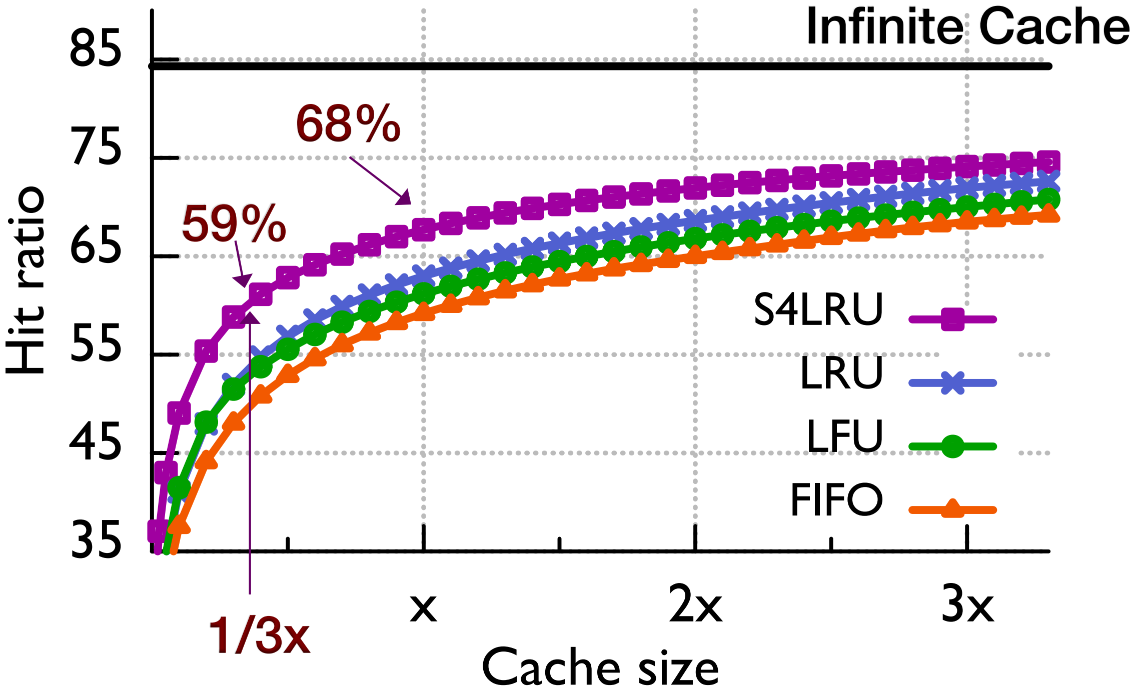
- "Infinite" size ratio needs 45x of capacity

Edge Cache with Different Algos



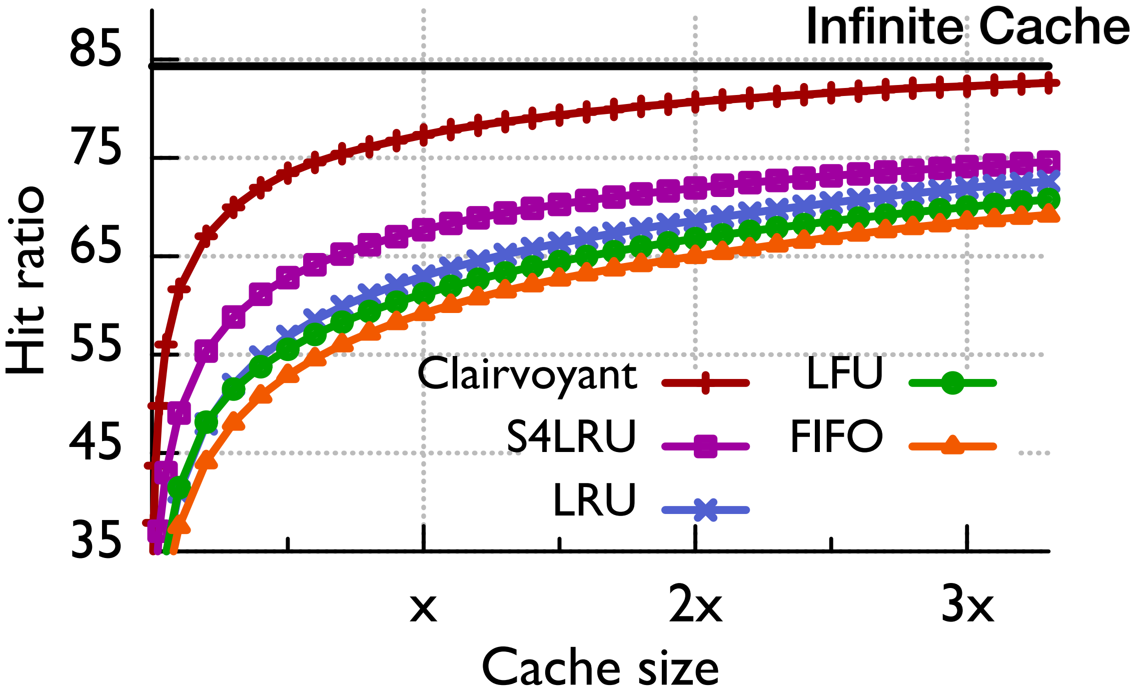
- LRU > LFU > FIFO

Edge Cache with Different Algos



- **S4LRU** is a more complex algorithm, uses recency and frequency

Edge Cache with Different Algos

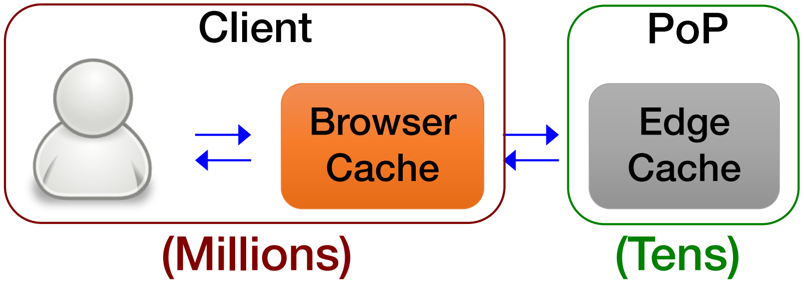


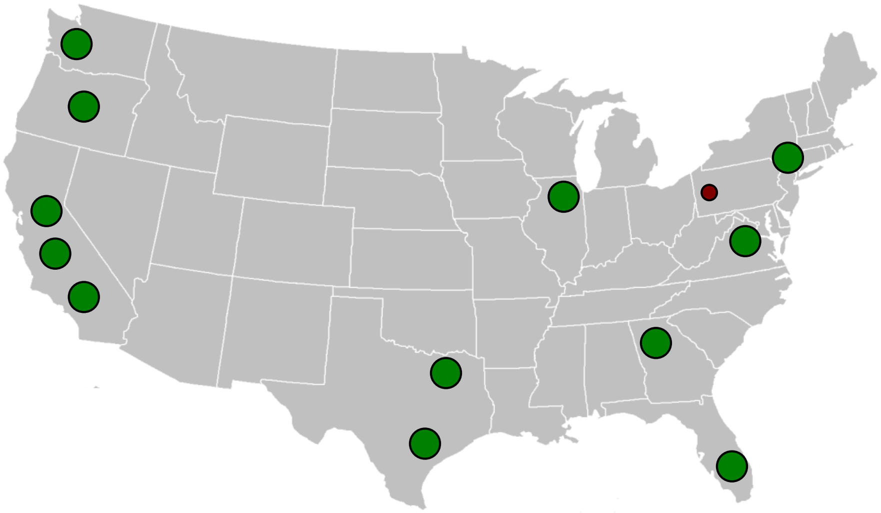
- Clairvoyant (Bélády) shows we can do much better!

A Hierarchy of CDN Caches

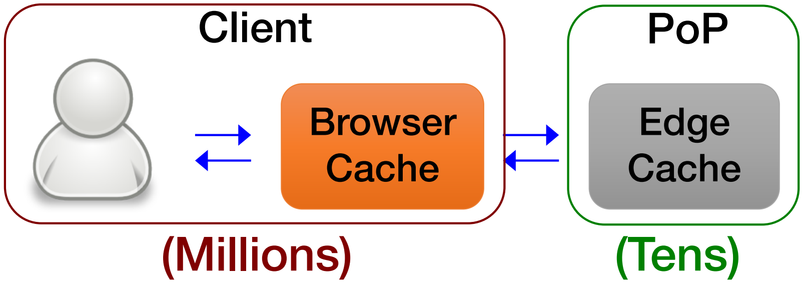
[Figures from Qi Huang's 2013 SOSP Talk]

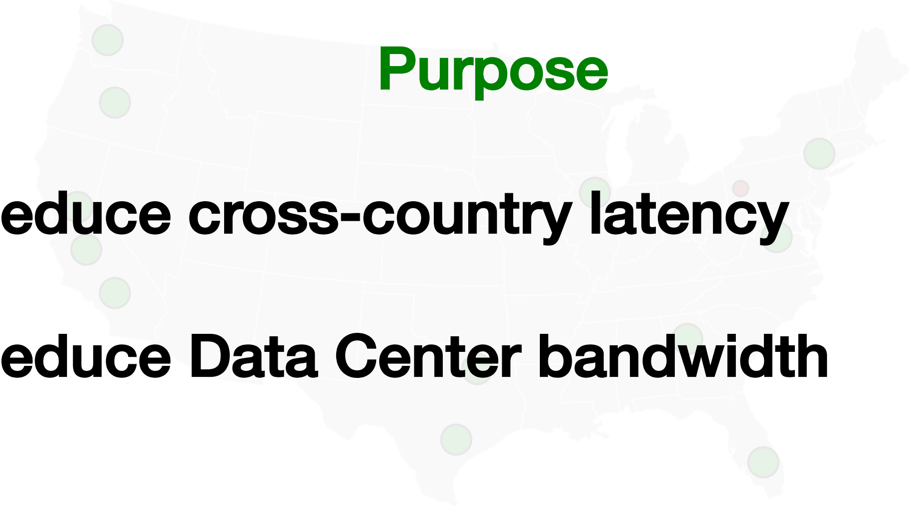
Geo-distributed Edge Cache (FIFO)





Geo-distributed Edge Cache (FIFO)

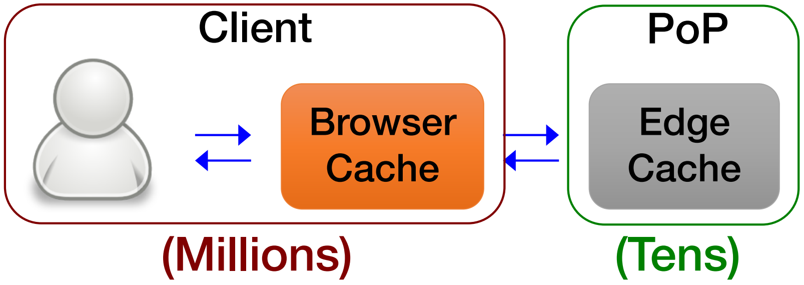




Purpose

1. **Reduce cross-country latency**
2. **Reduce Data Center bandwidth**

Geo-distributed Edge Cache (FIFO)



Client

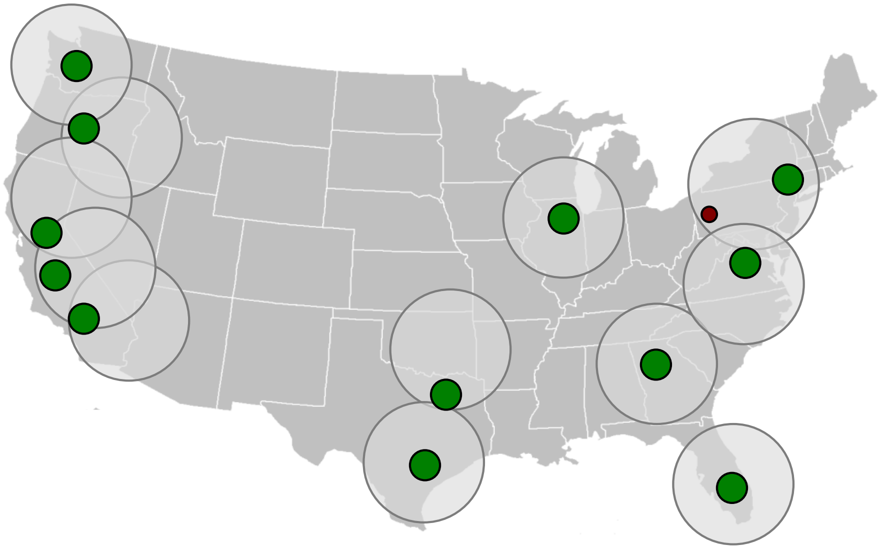
Browser Cache

PoP

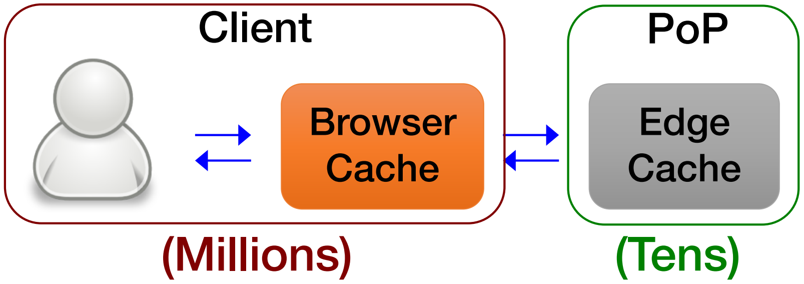
Edge Cache

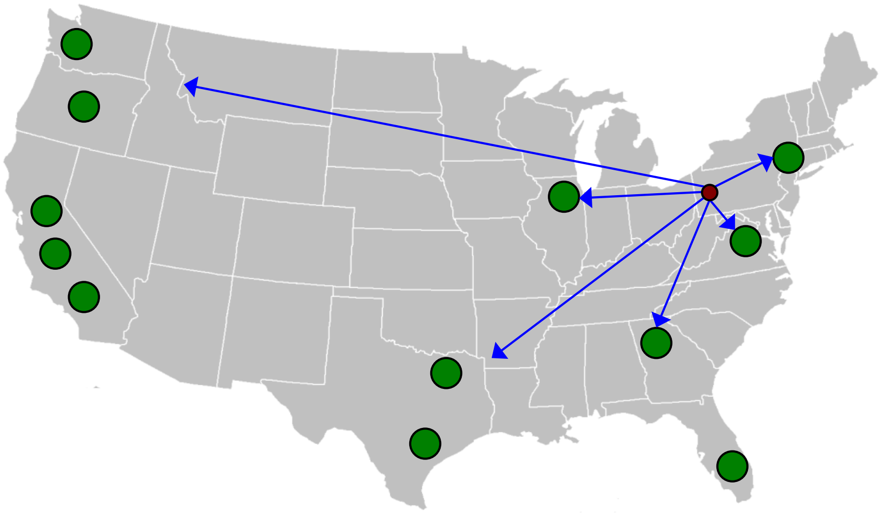
(Millions)

(Tens)

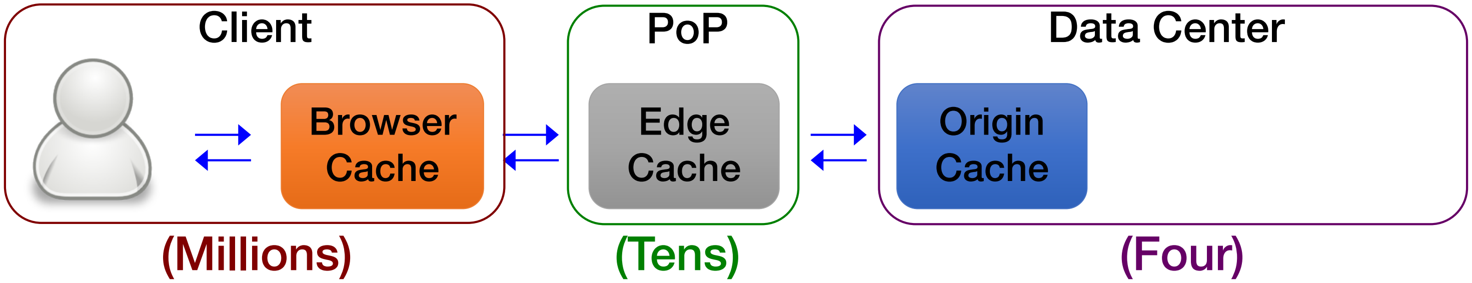


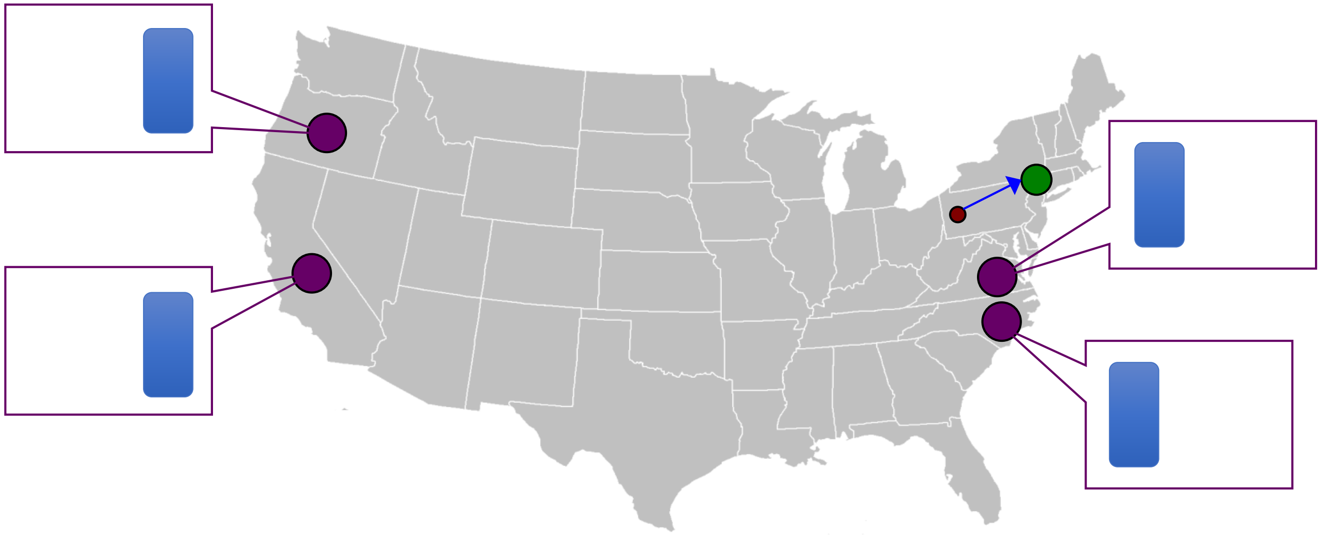
Geo-distributed Edge Cache (FIFO)



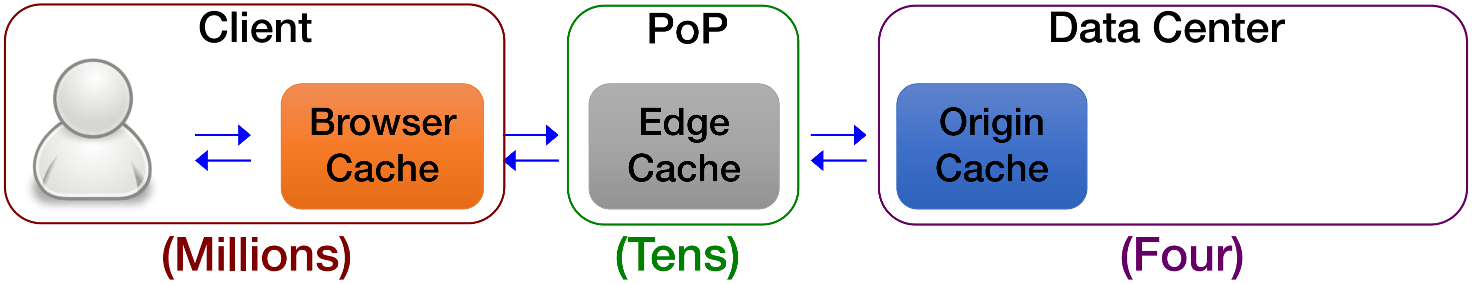


Single Global Origin Cache (FIFO)

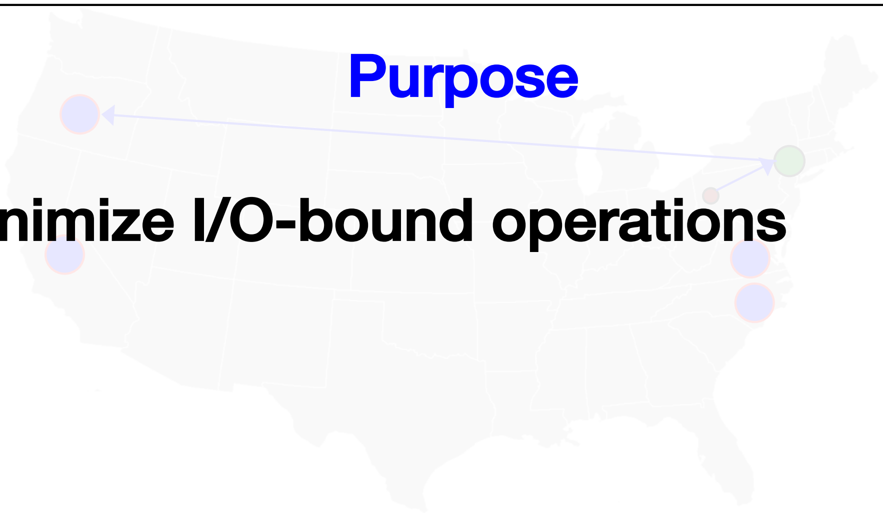




Single Global Origin Cache (FIFO)

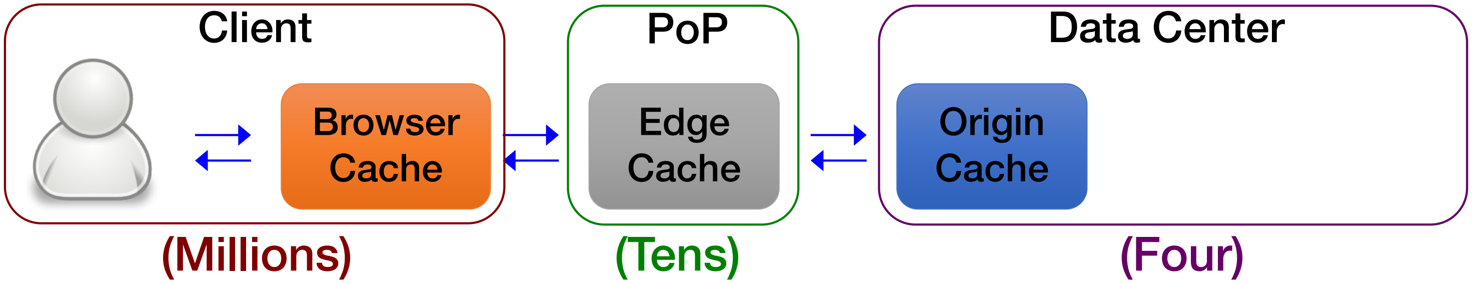


Purpose



1. Minimize I/O-bound operations

Single Global Origin Cache (FIFO)



Client

Browser Cache

(Millions)

PoP

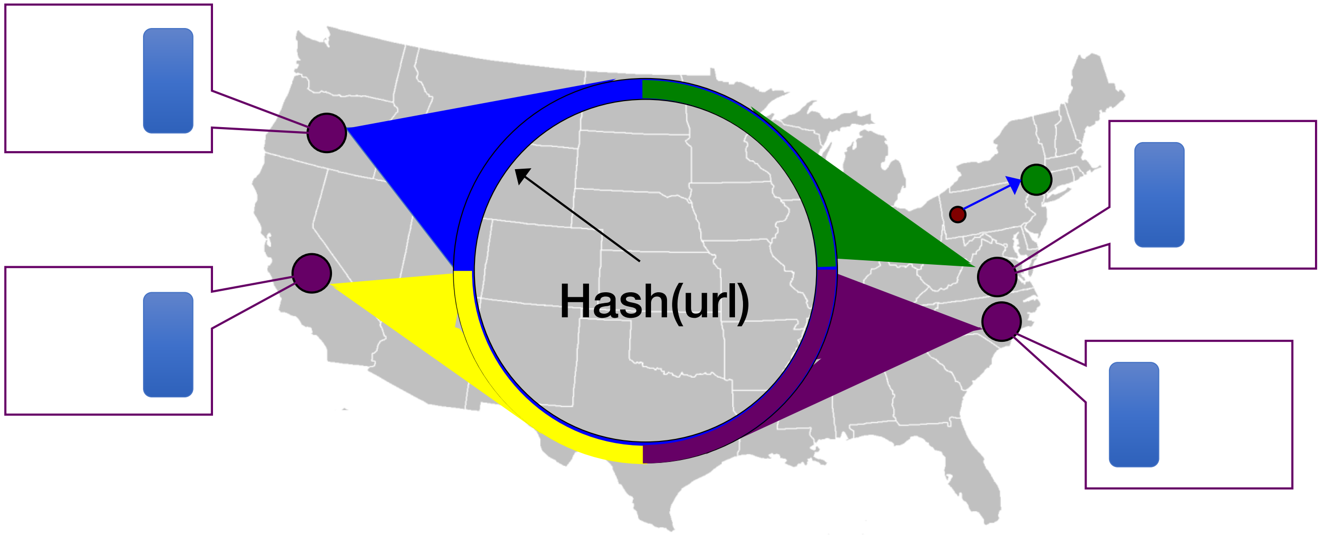
Edge Cache

(Tens)

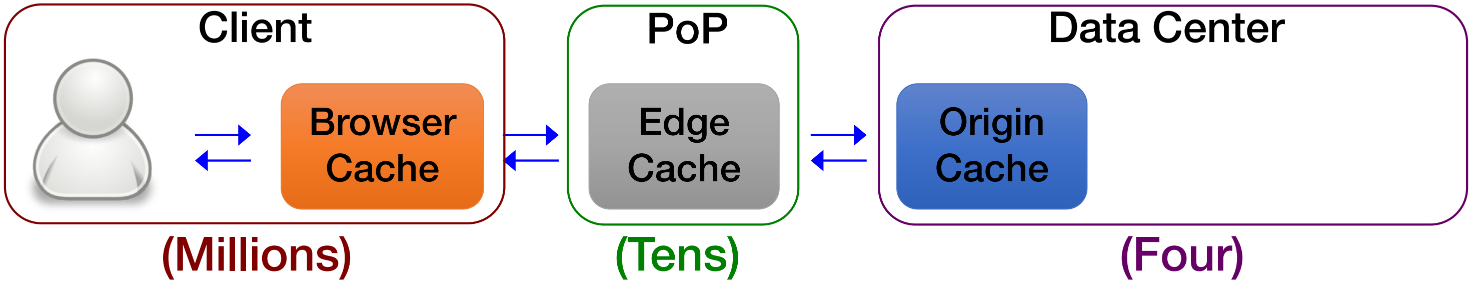
Data Center

Origin Cache

(Four)



Single Global Origin Cache (FIFO)



Client

Browser Cache

(Millions)

PoP

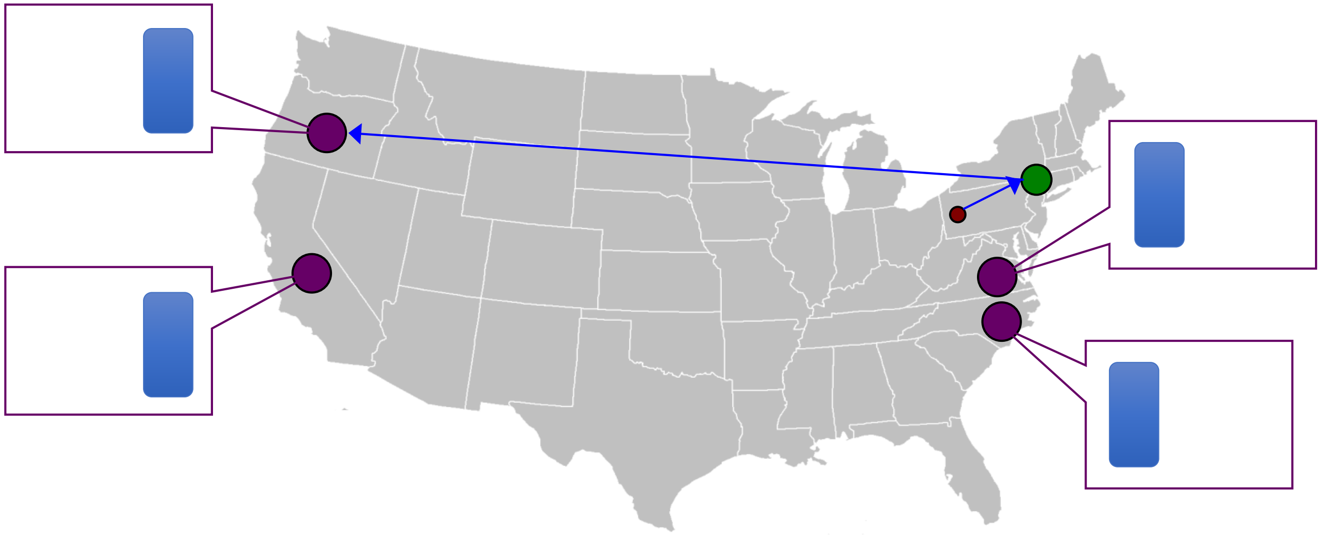
Edge Cache

(Tens)

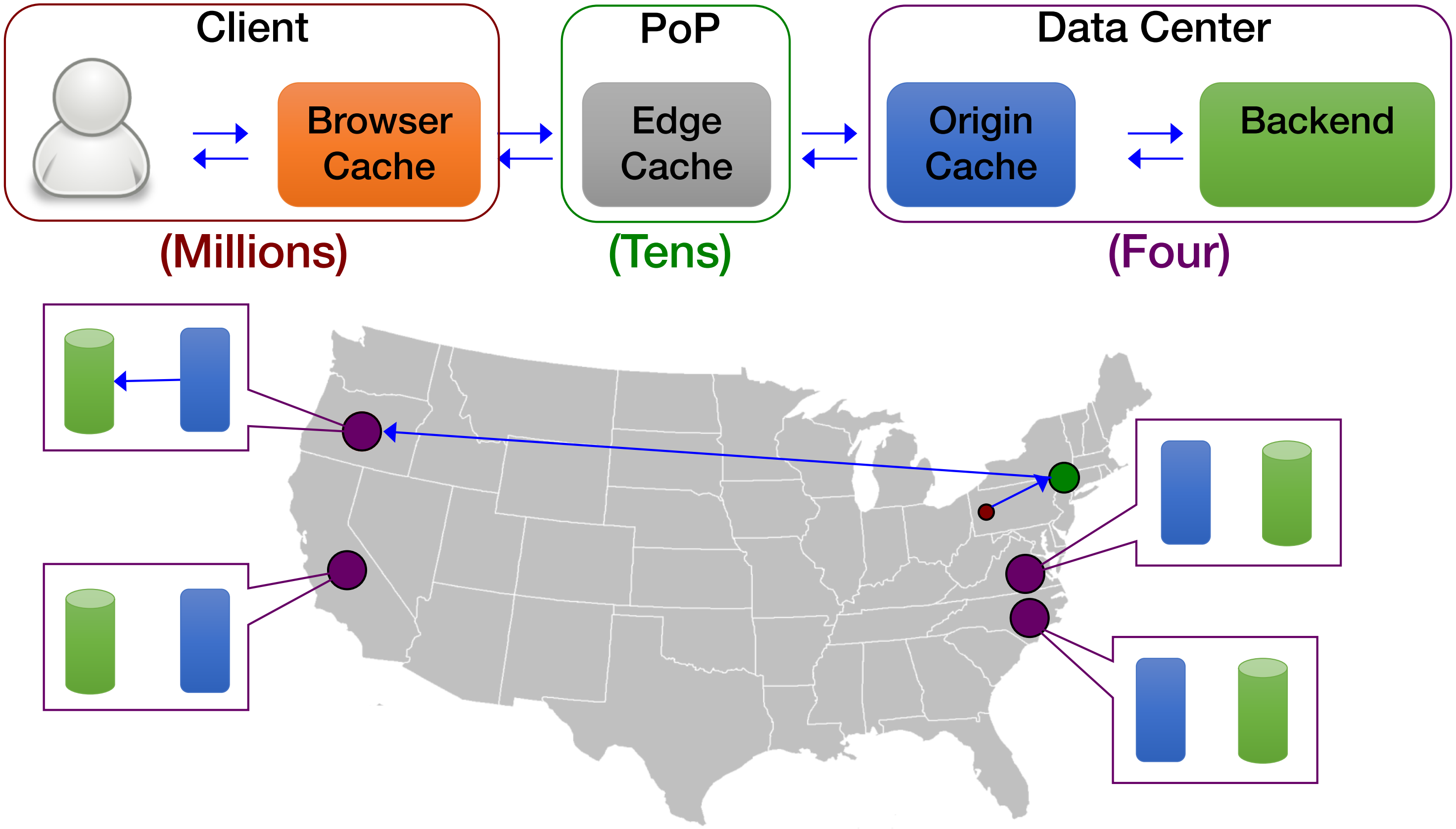
Data Center

Origin Cache

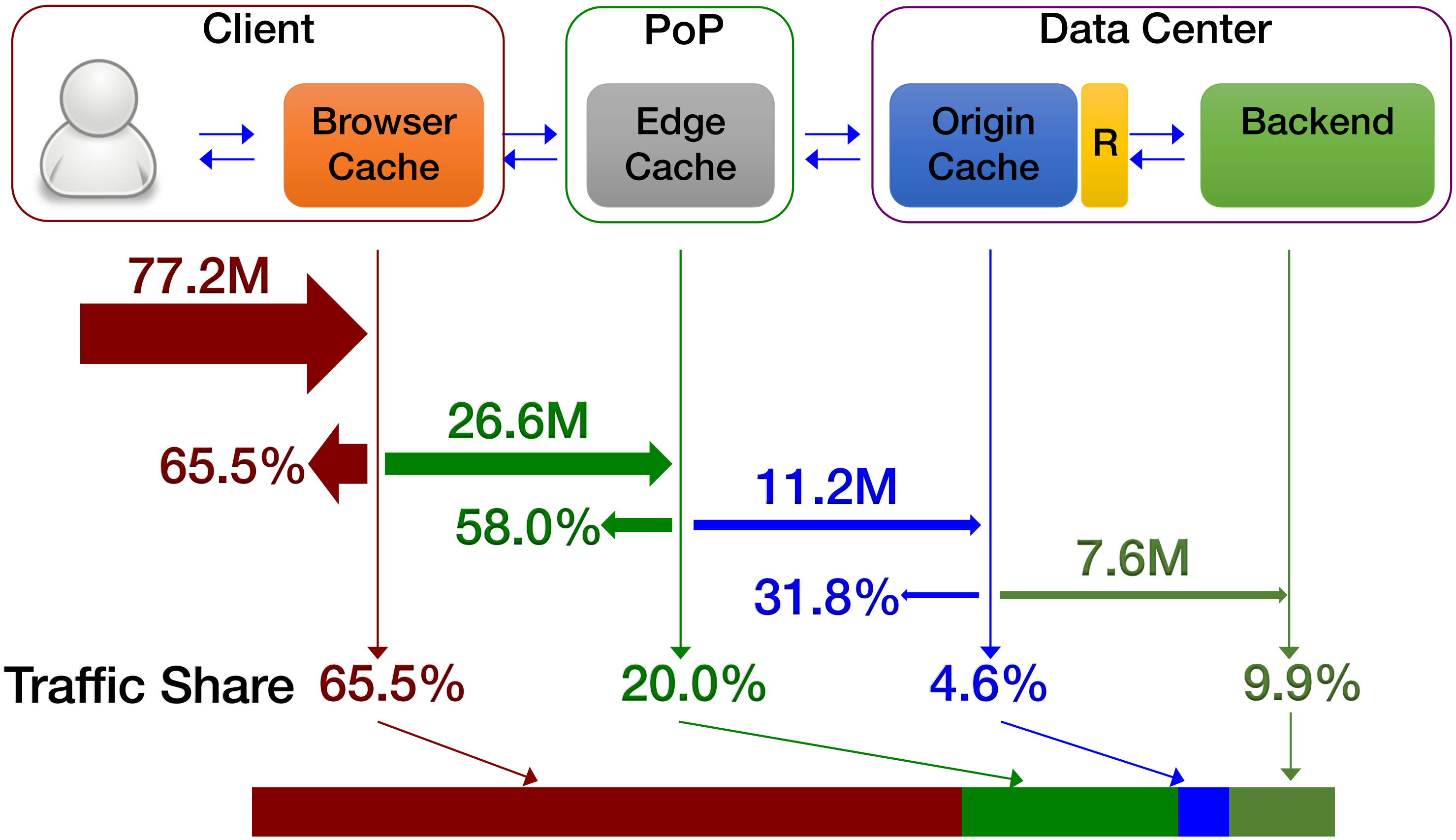
(Four)



Backend


Client
Browser Cache
(Millions)
PoP
Edge Cache
(Tens)
Data Center
Origin Cache
Backend
(Four)

CDN Effectiveness



Content Distribution Networks

- Serve “objects” in web pages, and much more, e.g., video segments

- Caching Algorithms: What do we store on them?
    - FIFO, LRU, LFU, Belady, … active area of research

- CDN Hierarchy
    - Smaller / more localized caches → bigger / more general caches