Content Distribution Networks (CDNs)

COS 418: Distributed Systems
Lecture 21

Wyatt Lloyd

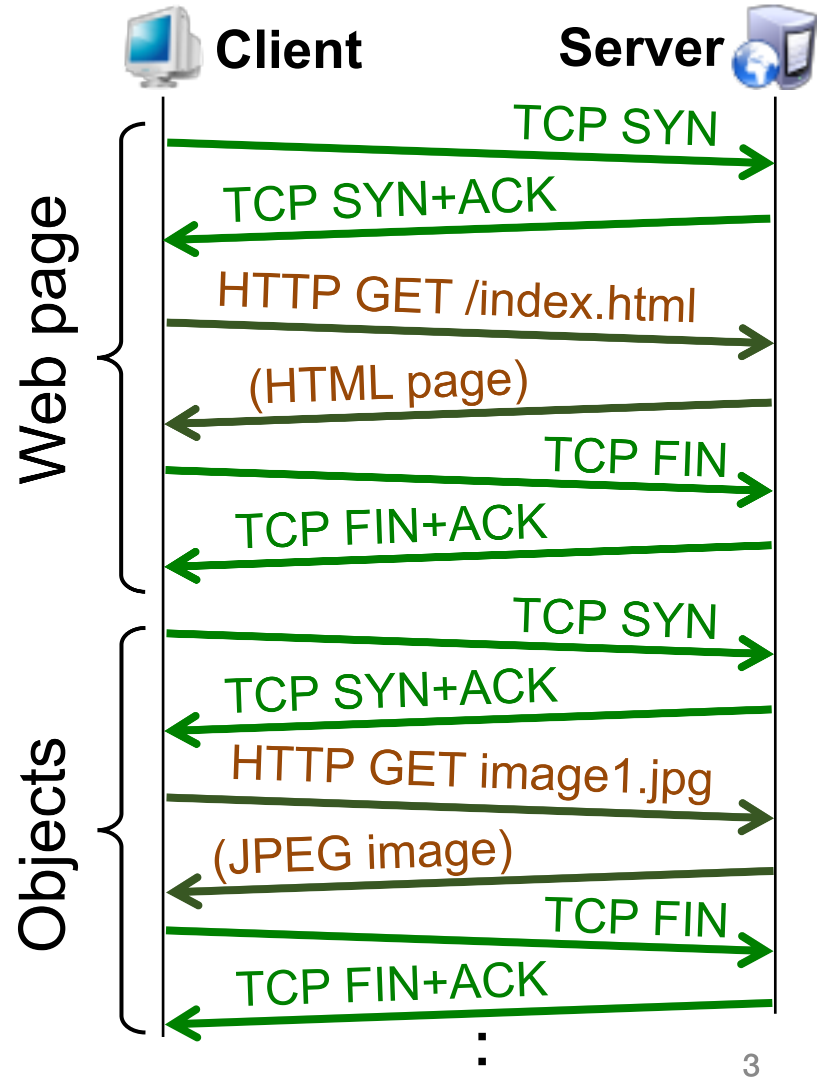
Content Distribution Networks

- Why & Overview
- Load Balancing
- Caching Algorithms
- Hierarchy

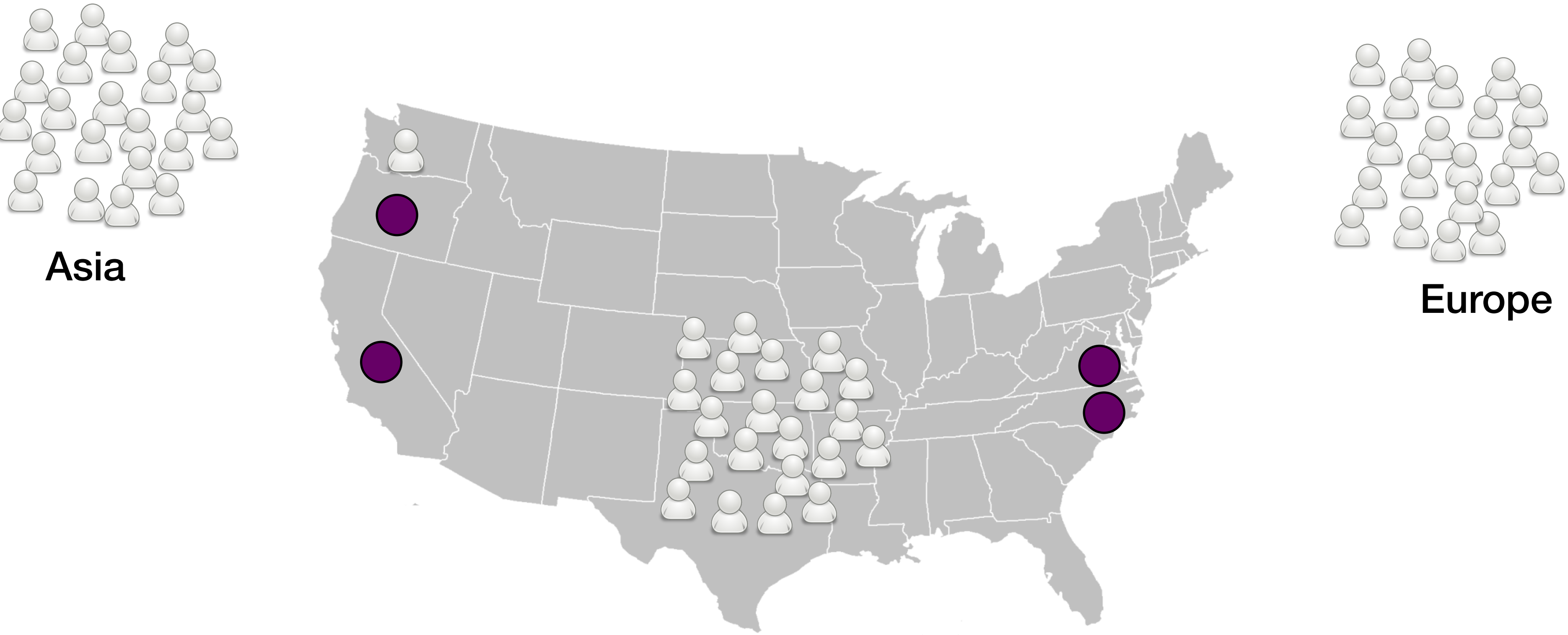
Anatomy of a Web Page Fetch

- Web page = HTML file + embedded images/objects

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

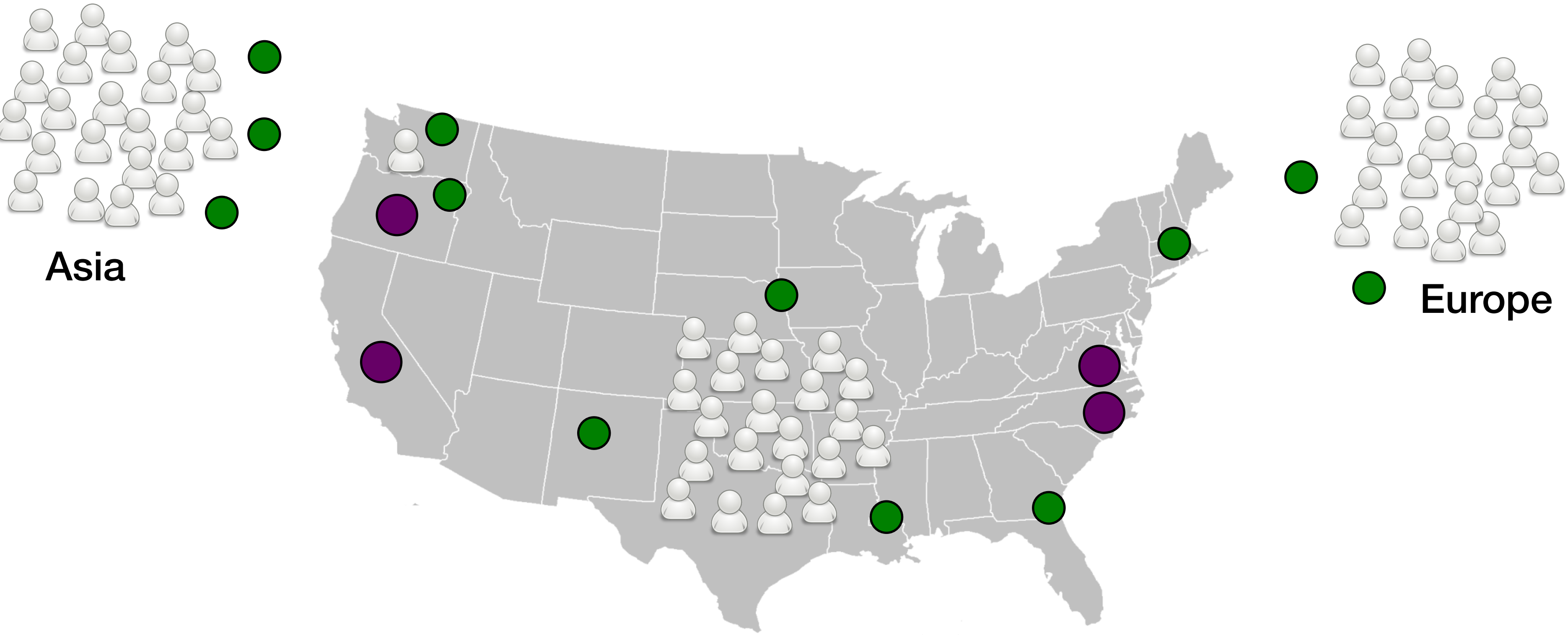


Serve It All From Datacenters?



- 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?

Content Distribution Networks

- Why & Overview
- Load Balancing
  - (Which CDN location do clients connect to?)
- Caching Algorithms
  - (What do we store on them?)
- Hierarchy

CDN Load Balancing

- Which of 1000+ options do I go to?

- Goals?
    - Nearby (lower latency)
    - Not overloaded (works)
    - (Cheaper bandwidth)
    - …

Load Balancing with IP Anycast

- IP Anycast
  - Multiple machines announce, “I am 1.2.3.4”
  - Internet routing sends packet to one of them
- Nearby?
  - Internet routing *often* sends packets to nearby machine!
  - (Take 461 in the spring to learn about BGP)
- Not overloaded?
  - Internet routing has no idea…
- Microsoft’s First-Party CDN (Bing, Xbox) uses it
  - (With additional cleverness to deal with overload)

Load Balancing with DNS

- DNS: Global, distributed, eventually consistent database that maps names to IP addresses
  - E.g., “cs.princeton.edu” -> “128.112.136.35”
- Give clients urls, then regularly update mapping in DNS from urls to IP addresses
  - Detailed elided, take 461 in the spring!
- Nearby?
  - DNS protocol *often* allows nearby matching
- Not overloaded?
  - Regularly updating mapping allows *pretty good* load balancing
    - e.g., If location X is heavily loaded, stop sending clients to it

Load Balancing with DC Control

- Datacenter based control: tell clients what CDN location to go to directly in url
  - e.g., ewr.cs.princeton.edu -> location near newark
- Nearby?
  - Yes, assuming you actually know where clients are
    - (true for DNS as well)
- Not overloaded?
  - Yes, fine-grained control per client

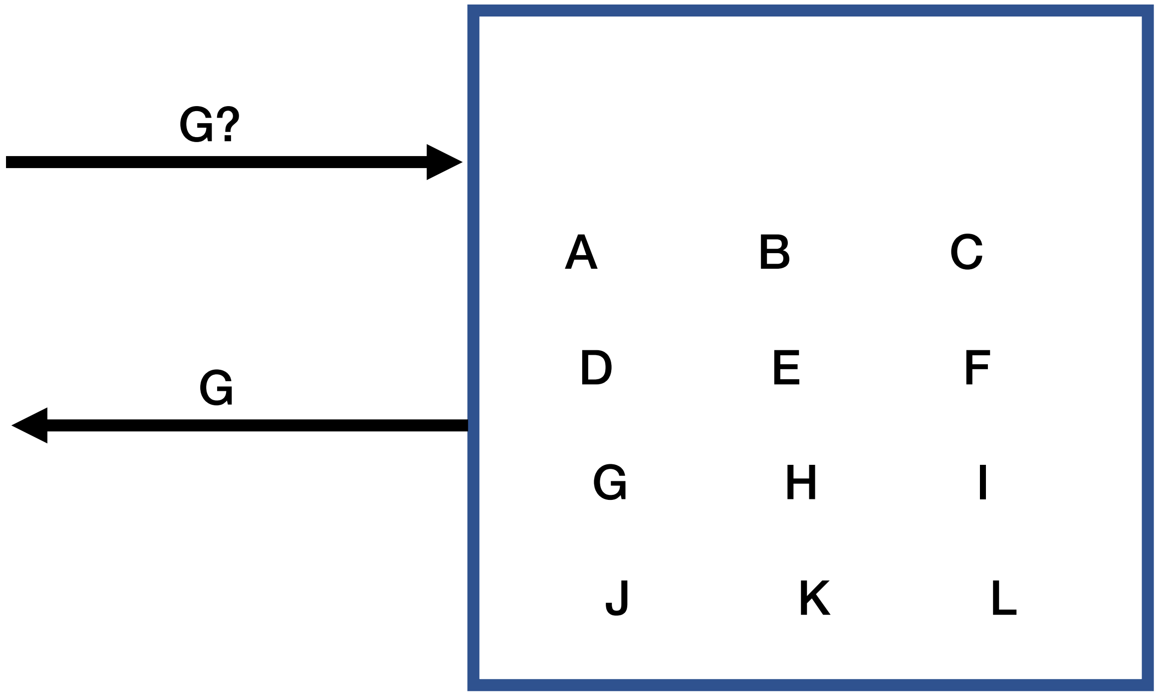
Content Distribution Networks

- Why & Overview
- Load Balancing
  - (Which CDN location do clients connect to?)
  - Several options, all of which are interesting distributed systems designed to work with internet routing
- Caching Algorithms
  - (What do we store on them?)
- Hierarchy

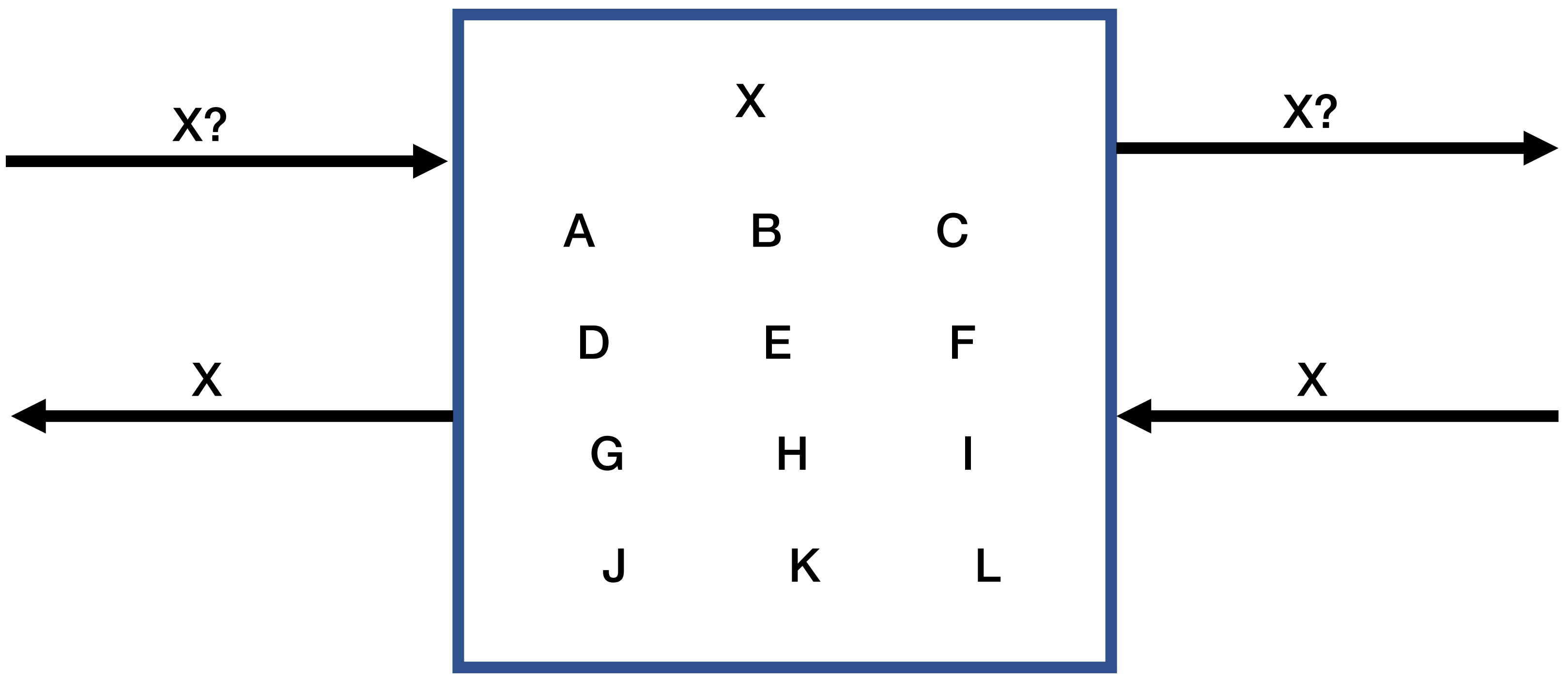
CDN Locations Store What?

- Store everything?
  - f4 at FB stored over 65PBs of photos/videos as of 2014[OSDI'14]
  - 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



CDN Cache Miss



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

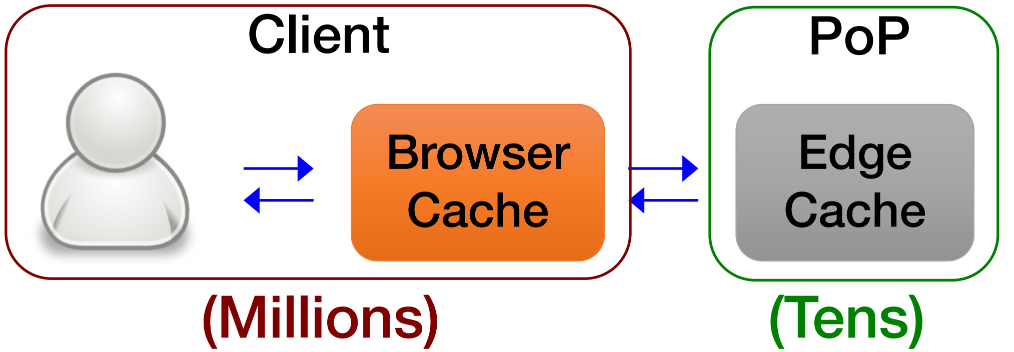
Cache Algorithms 101

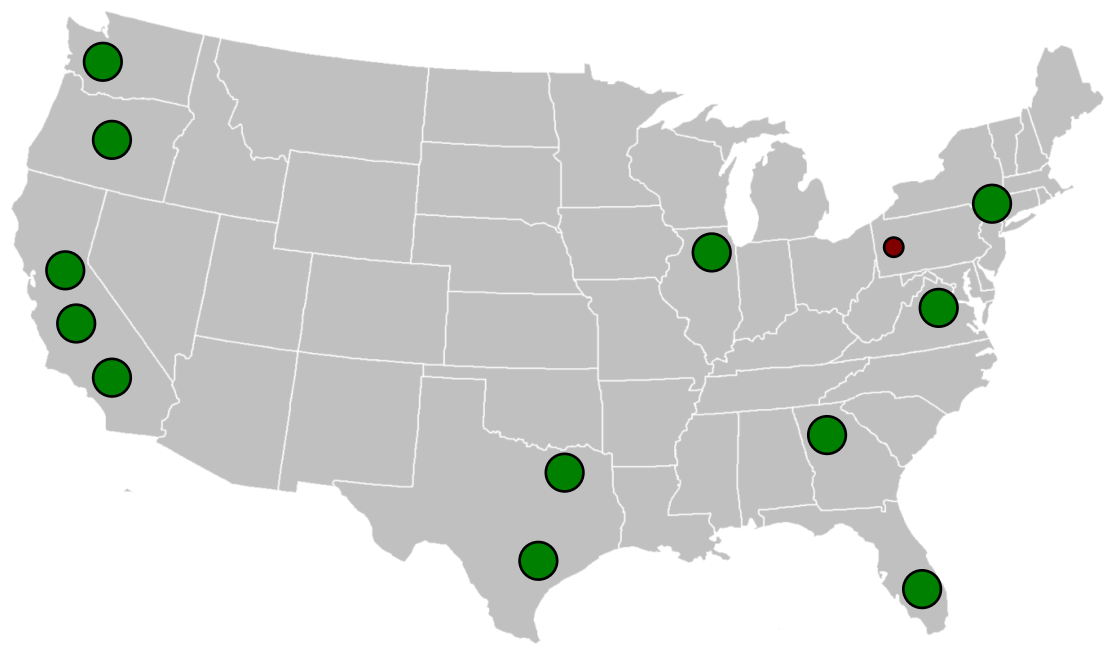
- First In First Out (FIFO)
  - Get rid of item put into the cache longest ago
- Least recently used (LRU)
  - Get rid of item in cache that was used longest ago
  - (Update access time on hit)
- Least frequently used (LFU)
  - Get rid of item in cache that was used the fewest number of time
  - (Update count on hit)

Content Distribution Networks

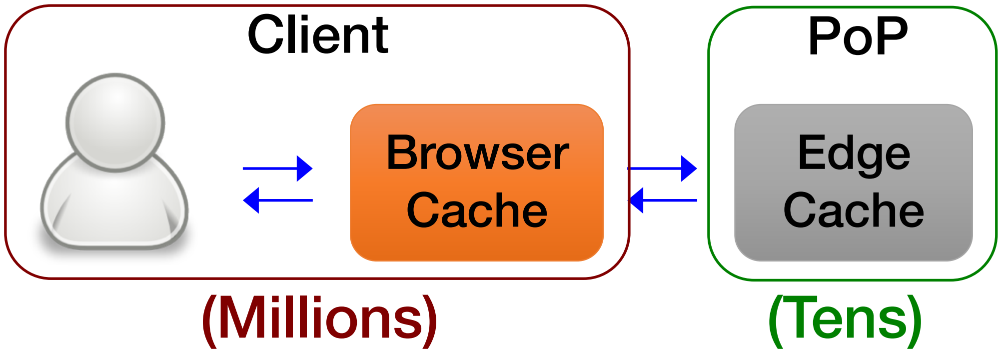
- Why & Overview
- Load Balancing
  - (Which CDN location do clients connect to?)
- Caching Algorithms
  - (What do we store on them?)
- Hierarchy
  - [Slides from Qi Huang's SOSP 2013 Talk]

Geo-distributed Edge Cache (FIFO)





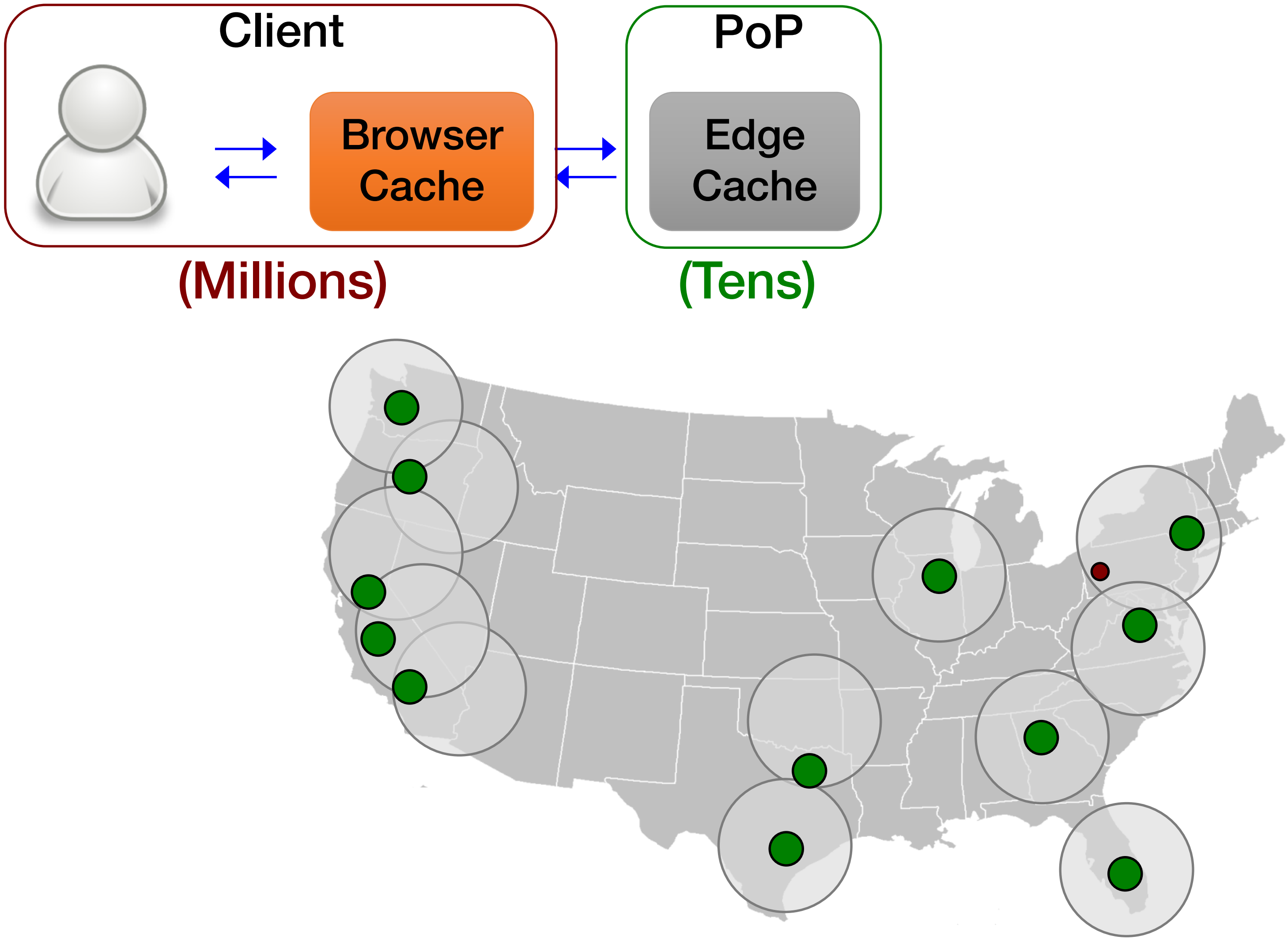
Geo-distributed Edge Cache (FIFO)



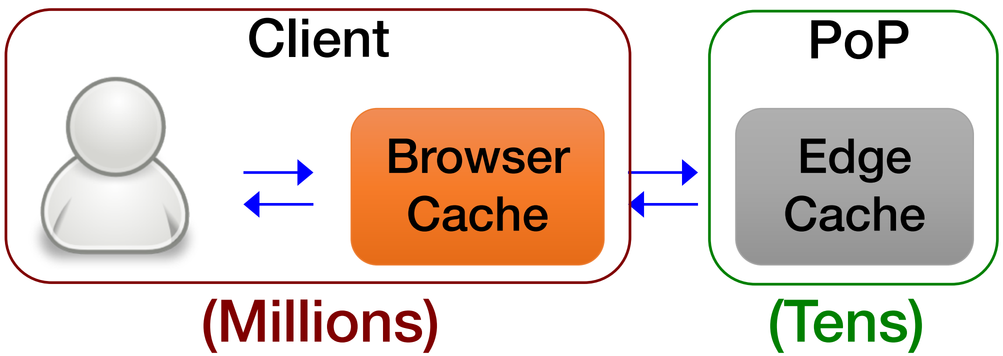
Purpose

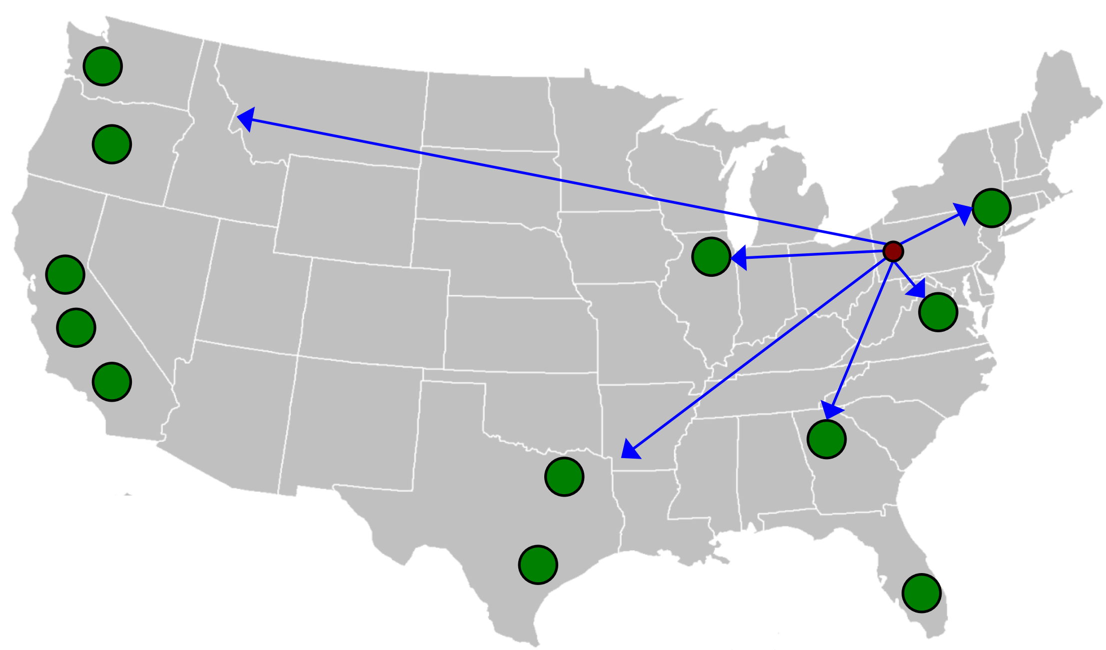
1. Reduce cross-country latency
2. Reduce Data Center bandwidth

Geo-distributed Edge Cache (FIFO)

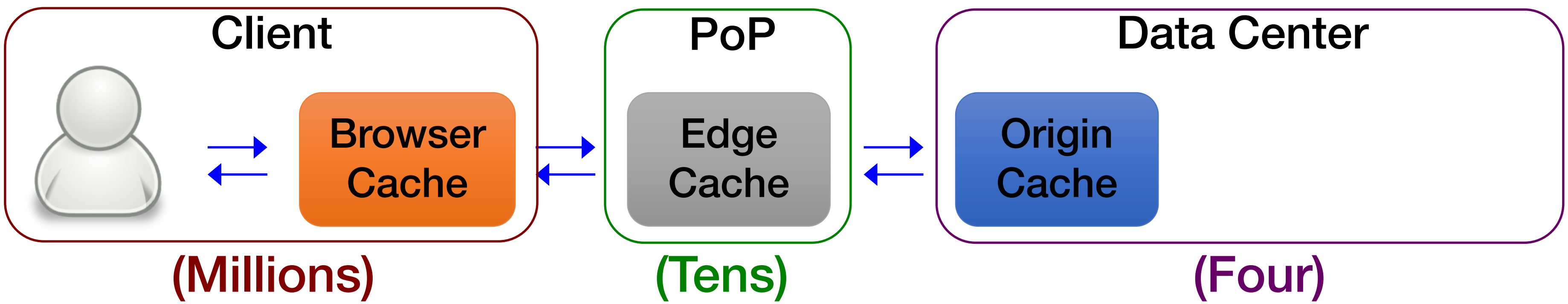


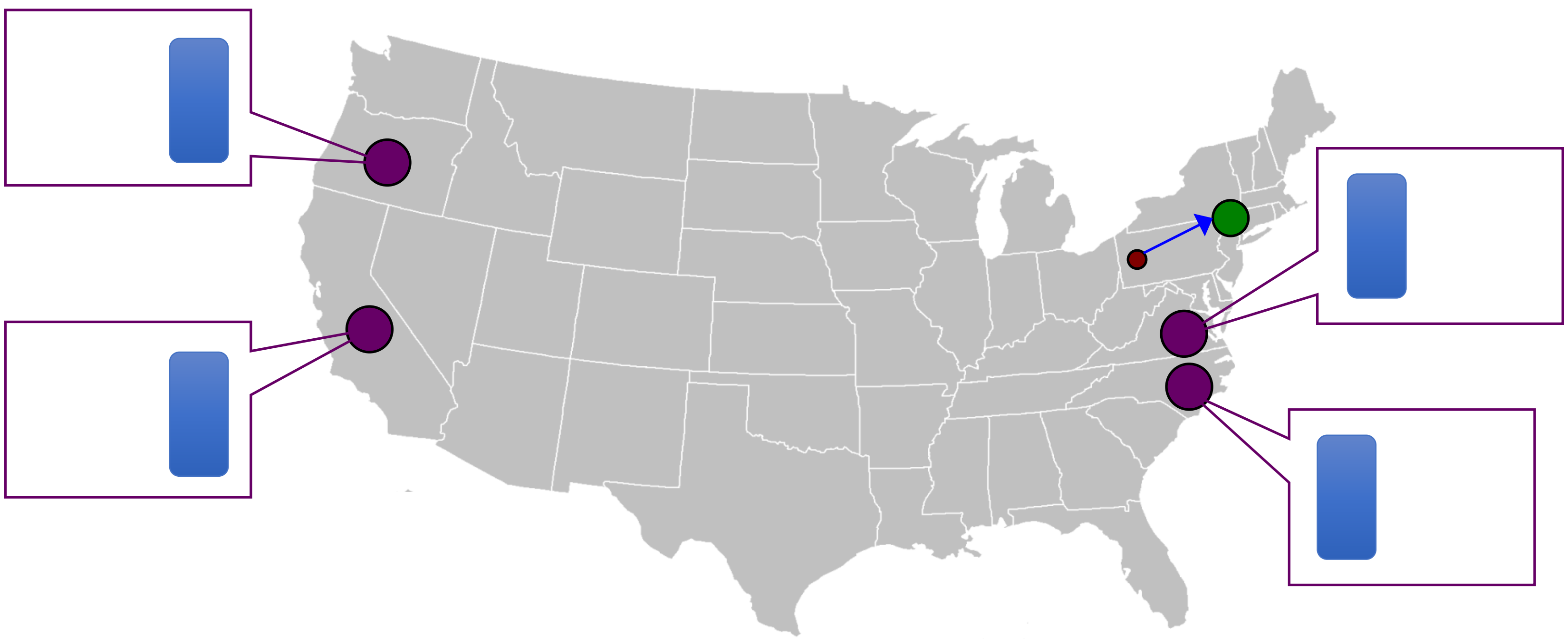
Geo-distributed Edge Cache (FIFO)



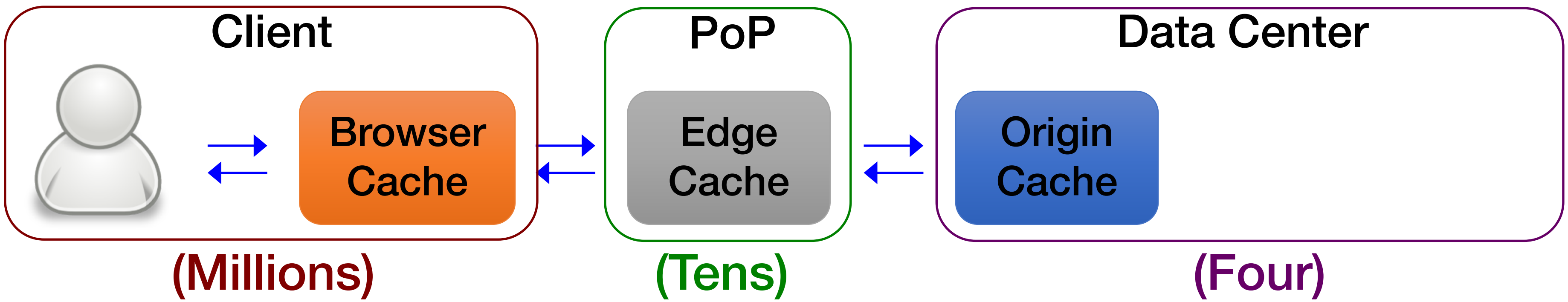


Single Global Origin Cache (FIFO)





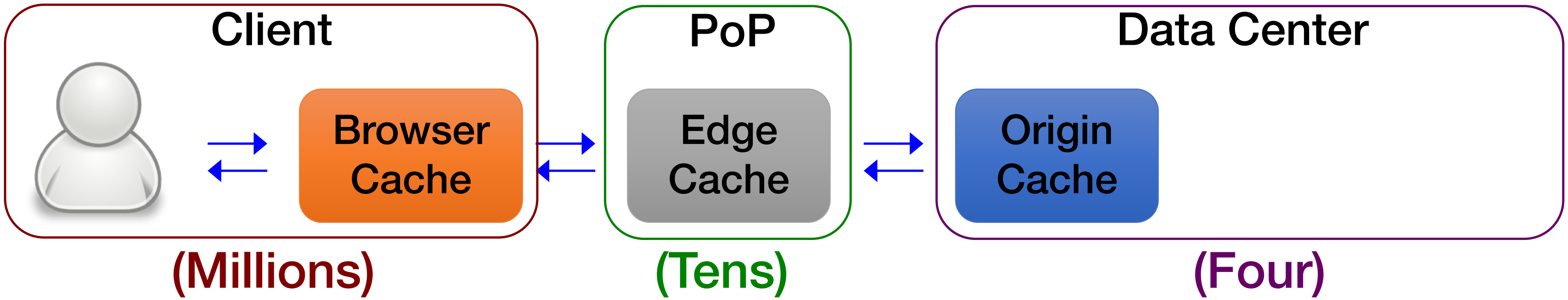
Single Global Origin Cache (FIFO)

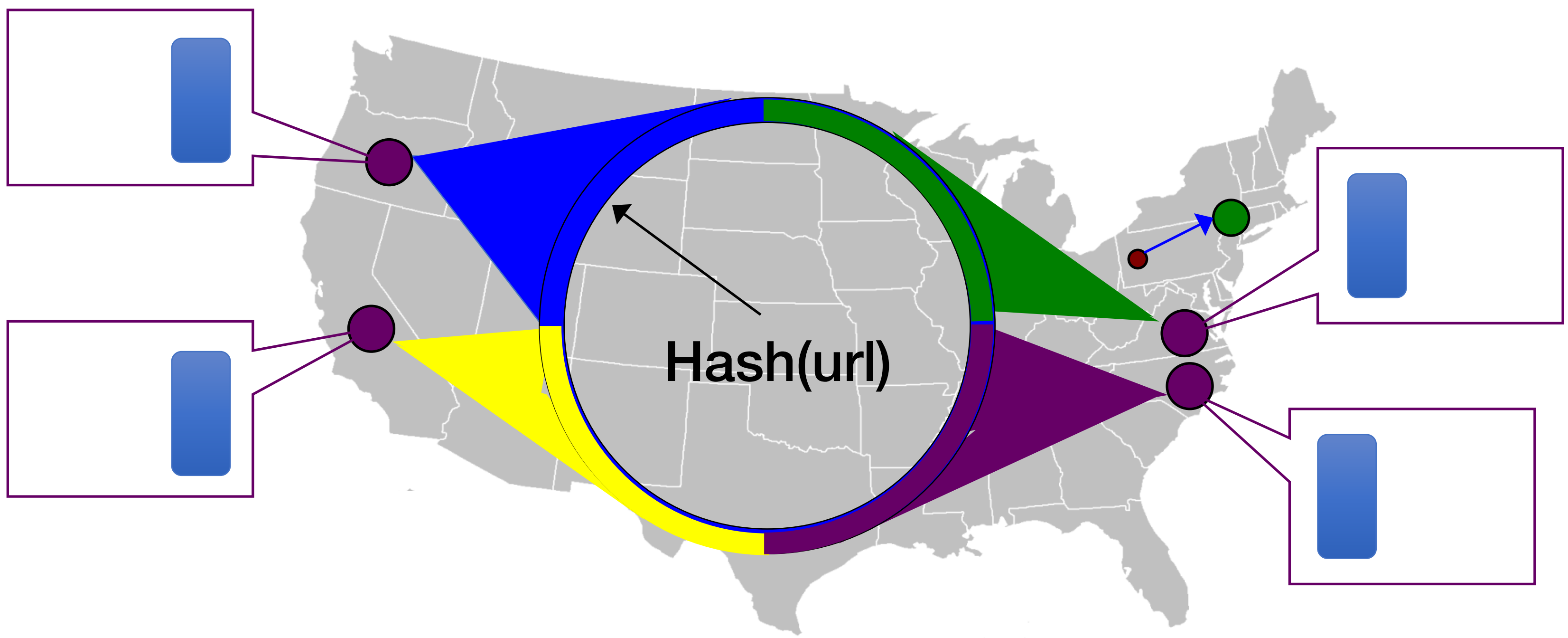


Purpose

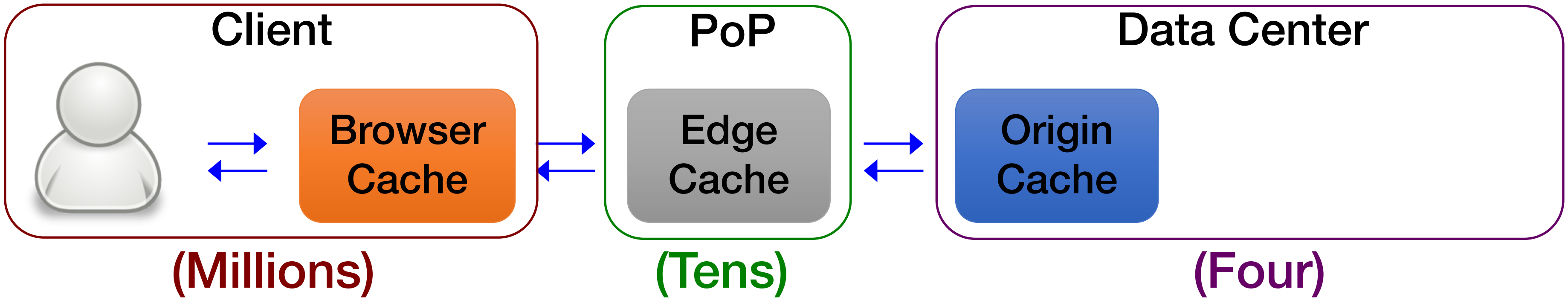
1. **Minimize I/O-bound operations**

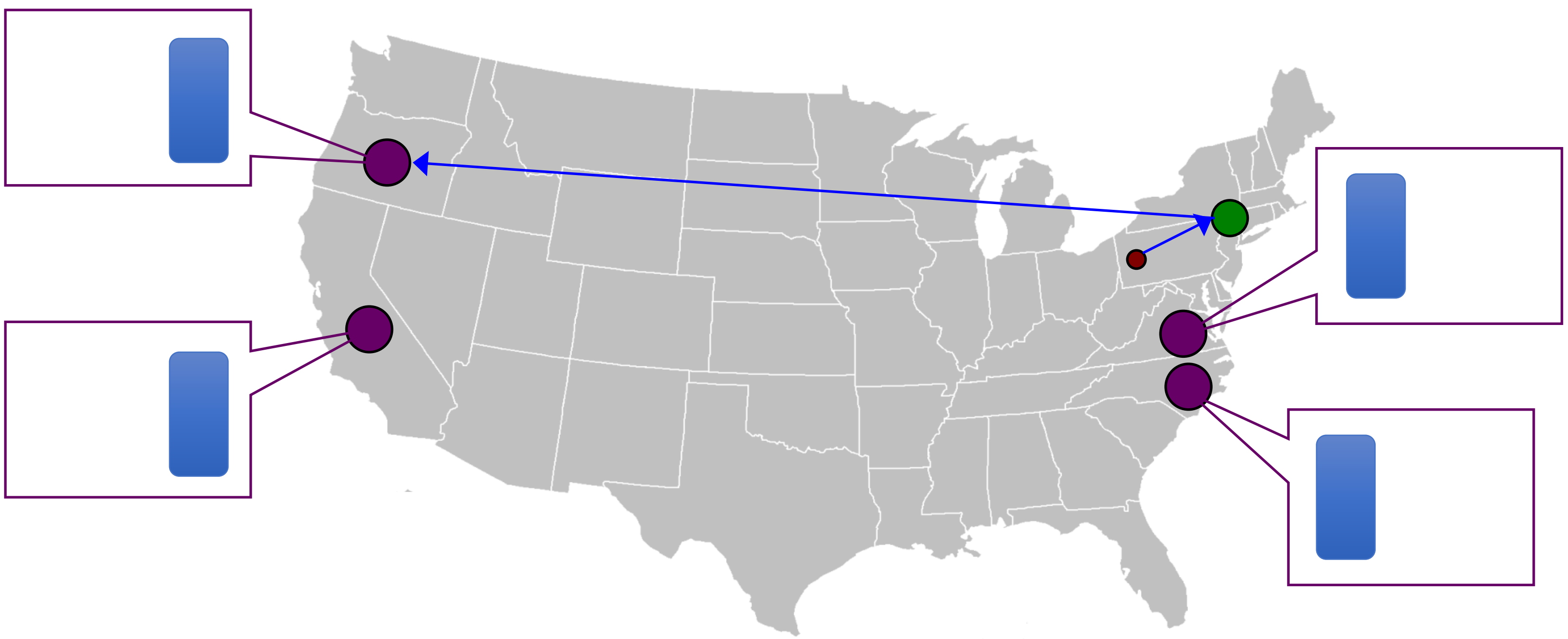
Single Global Origin Cache (FIFO)



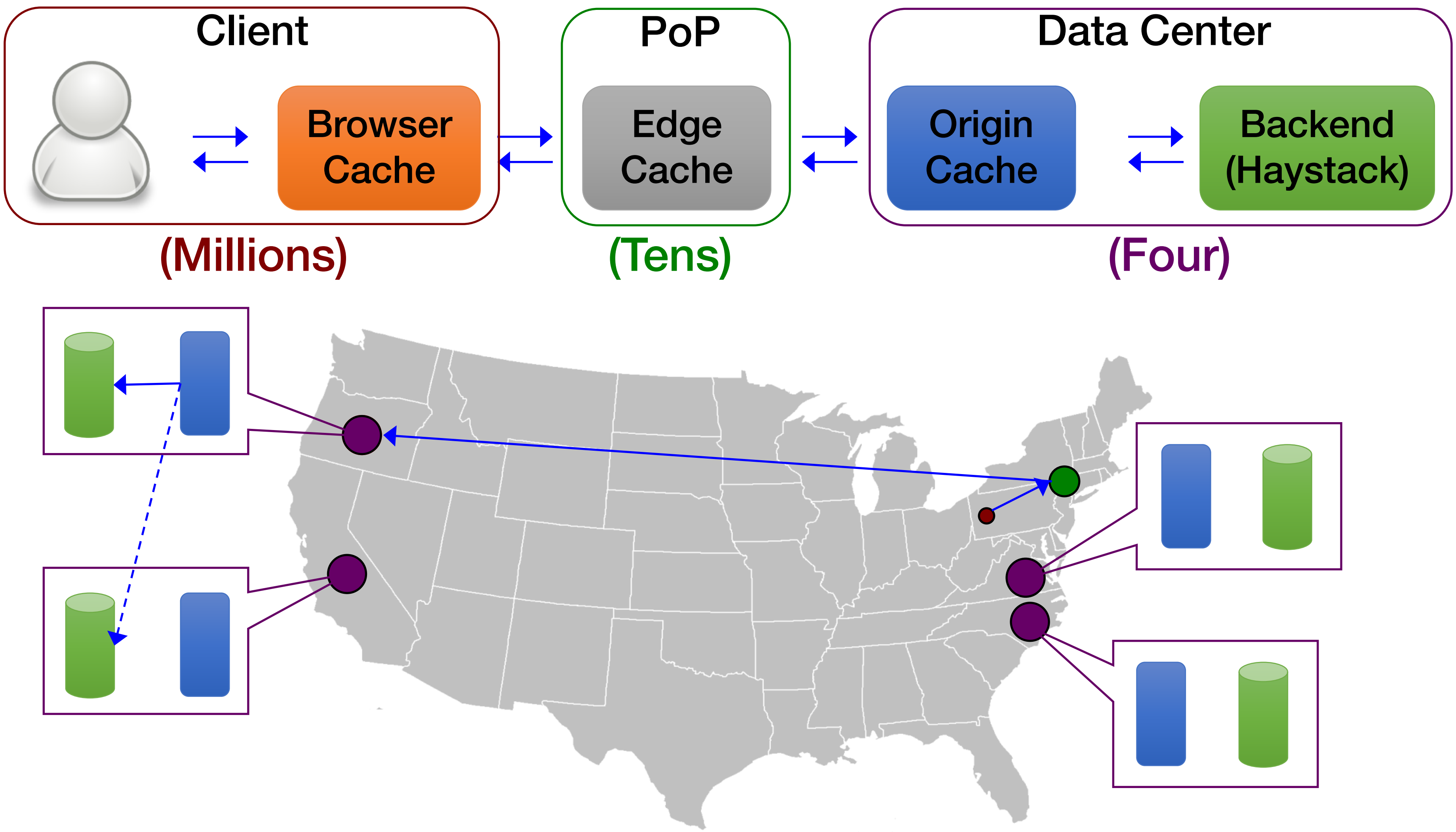


Single Global Origin Cache (FIFO)



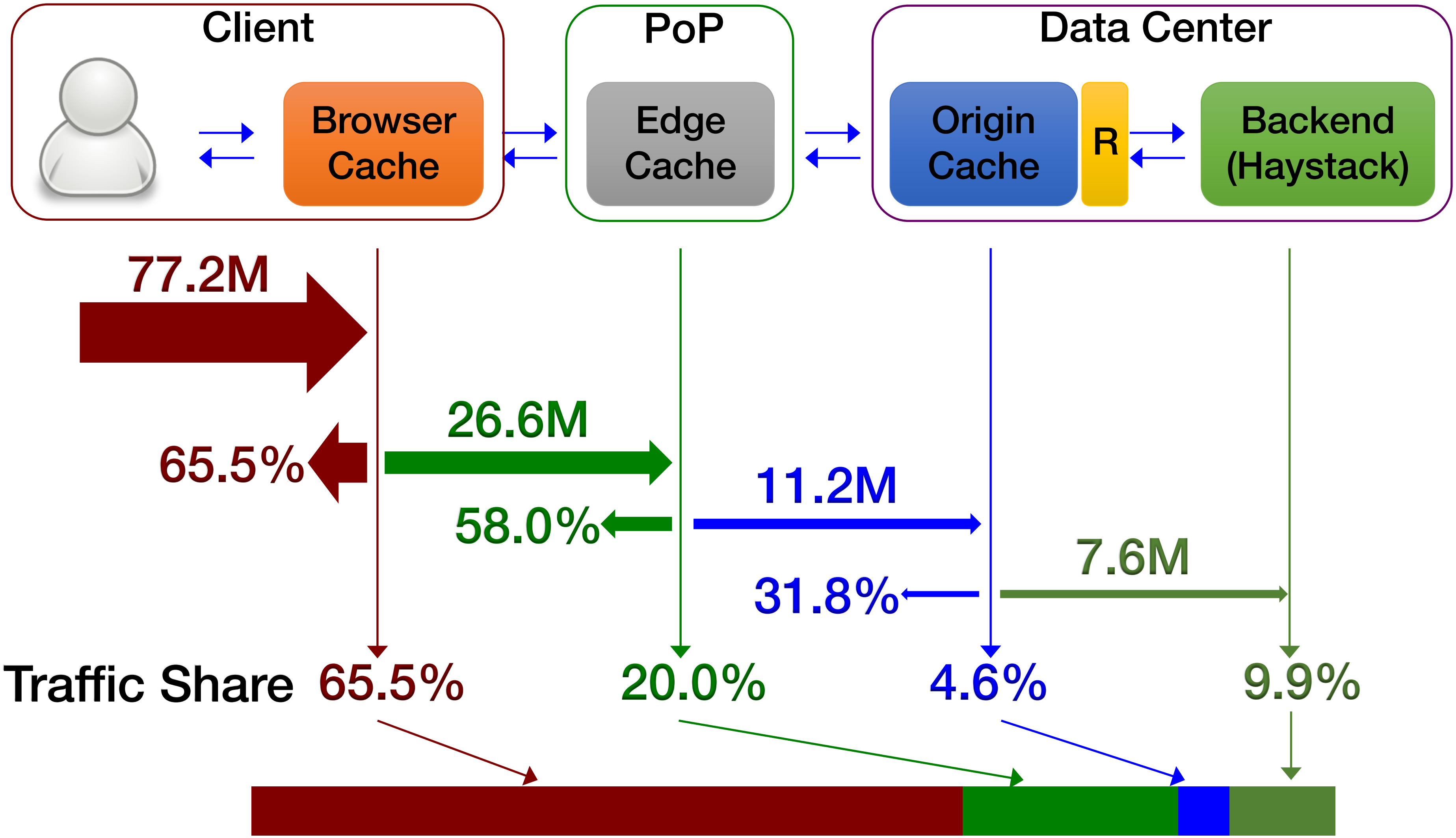


Haystack Backend



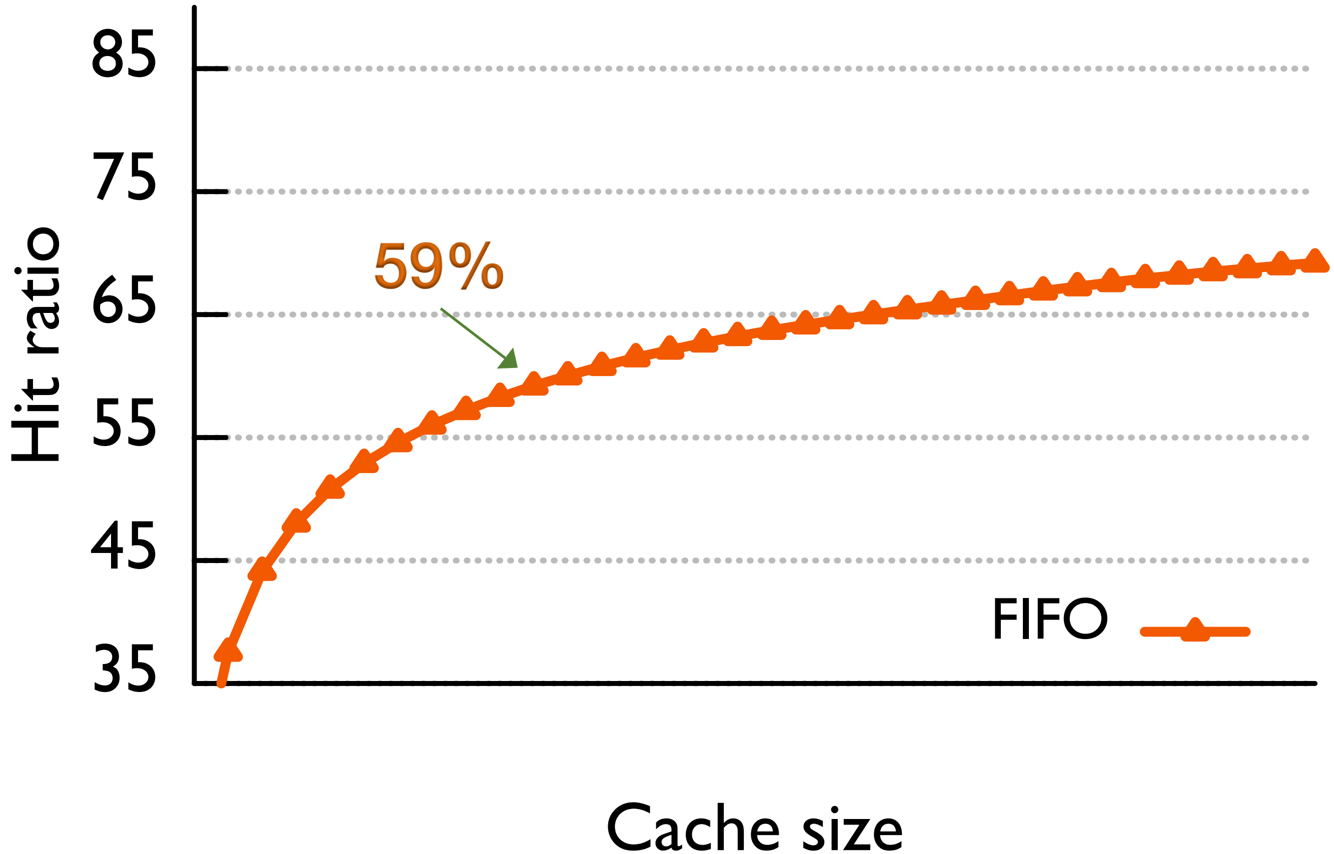
How Effective Was Facebook's CDN?

CDN Effectiveness [SOSP '13]



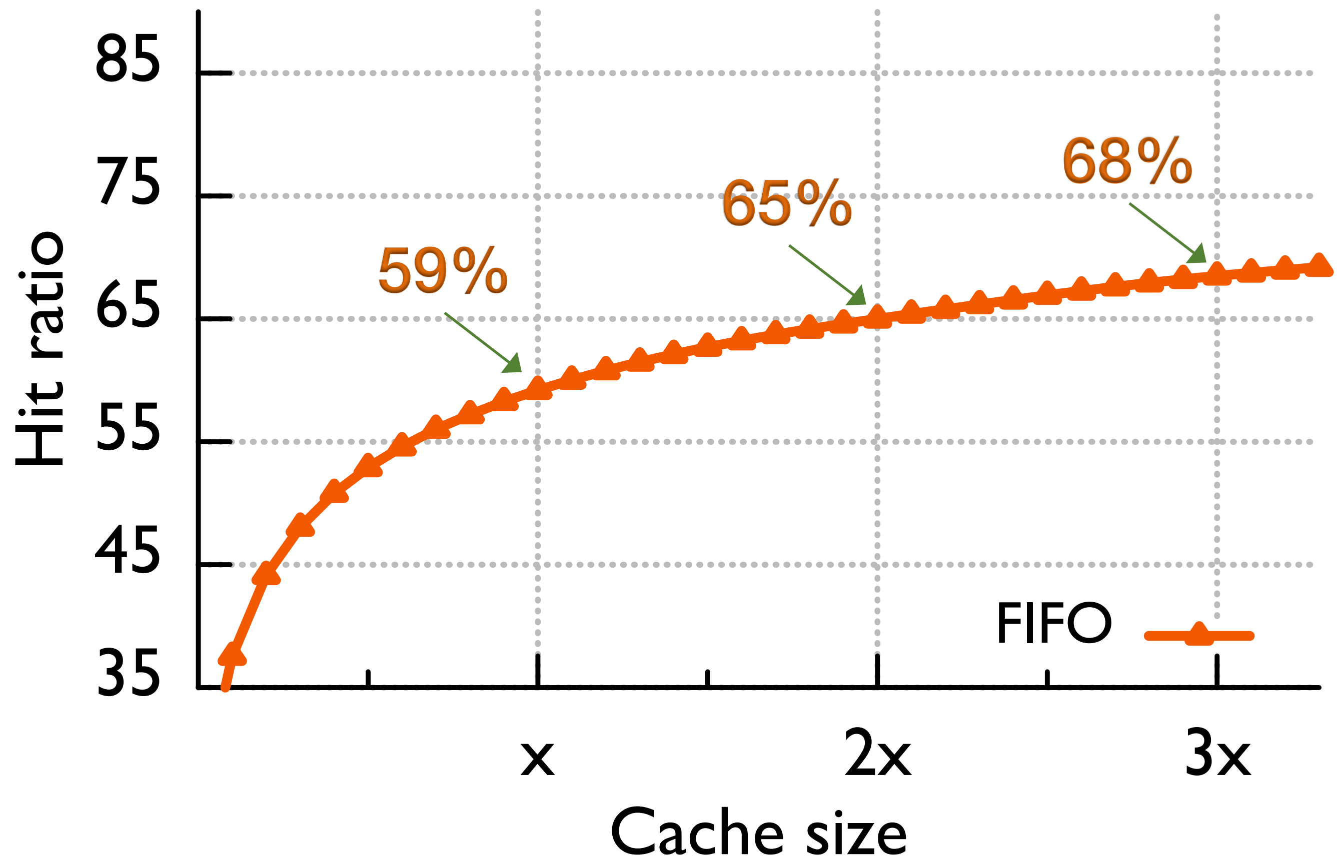
Can It Be Improved?

Edge Cache with Different Sizes



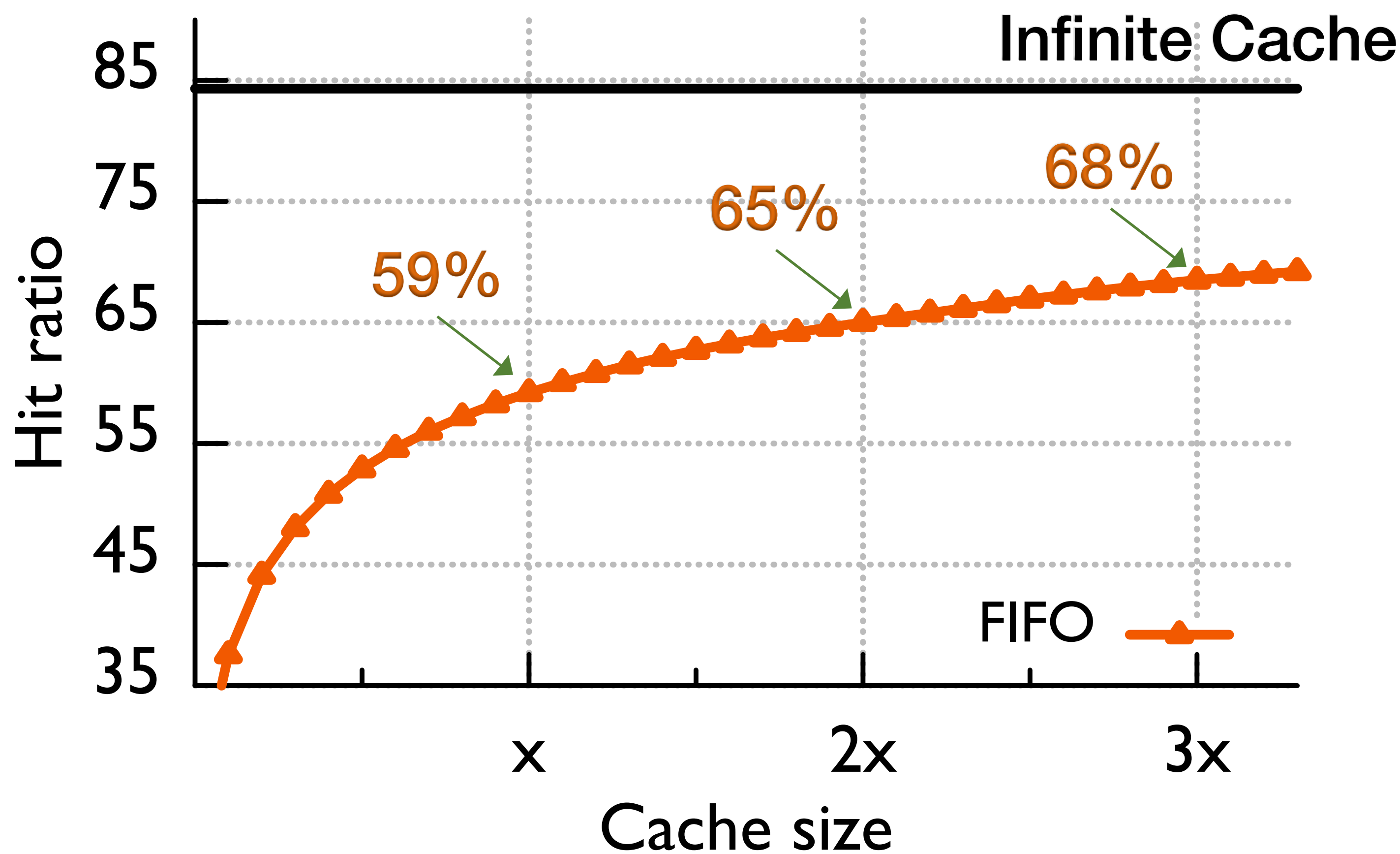
- Picked San Jose edge (high traffic, median hit ratio)

Edge Cache with Different Sizes



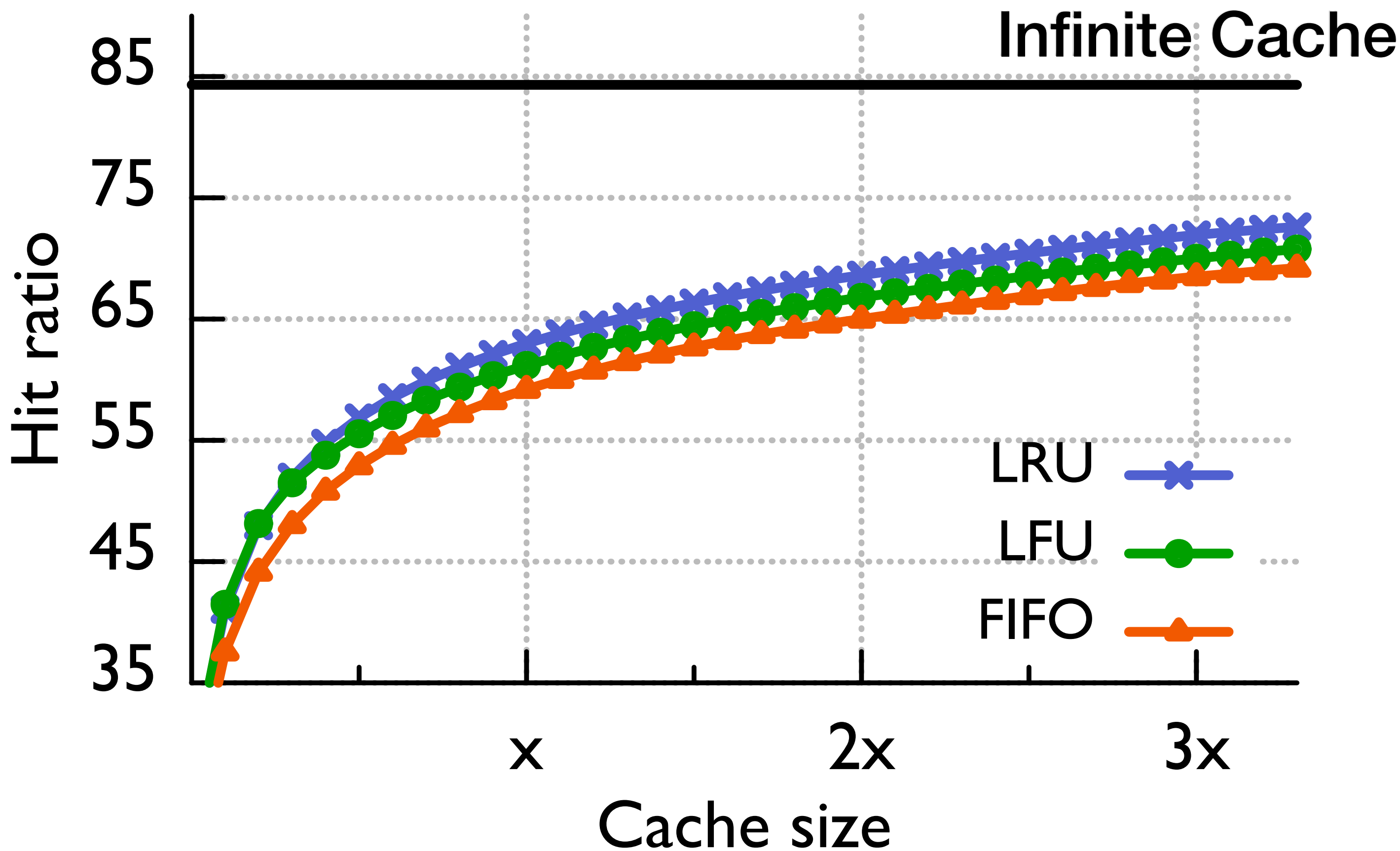
- "x" estimates current deployment size (59% hit ratio)

Edge Cache with Different Sizes



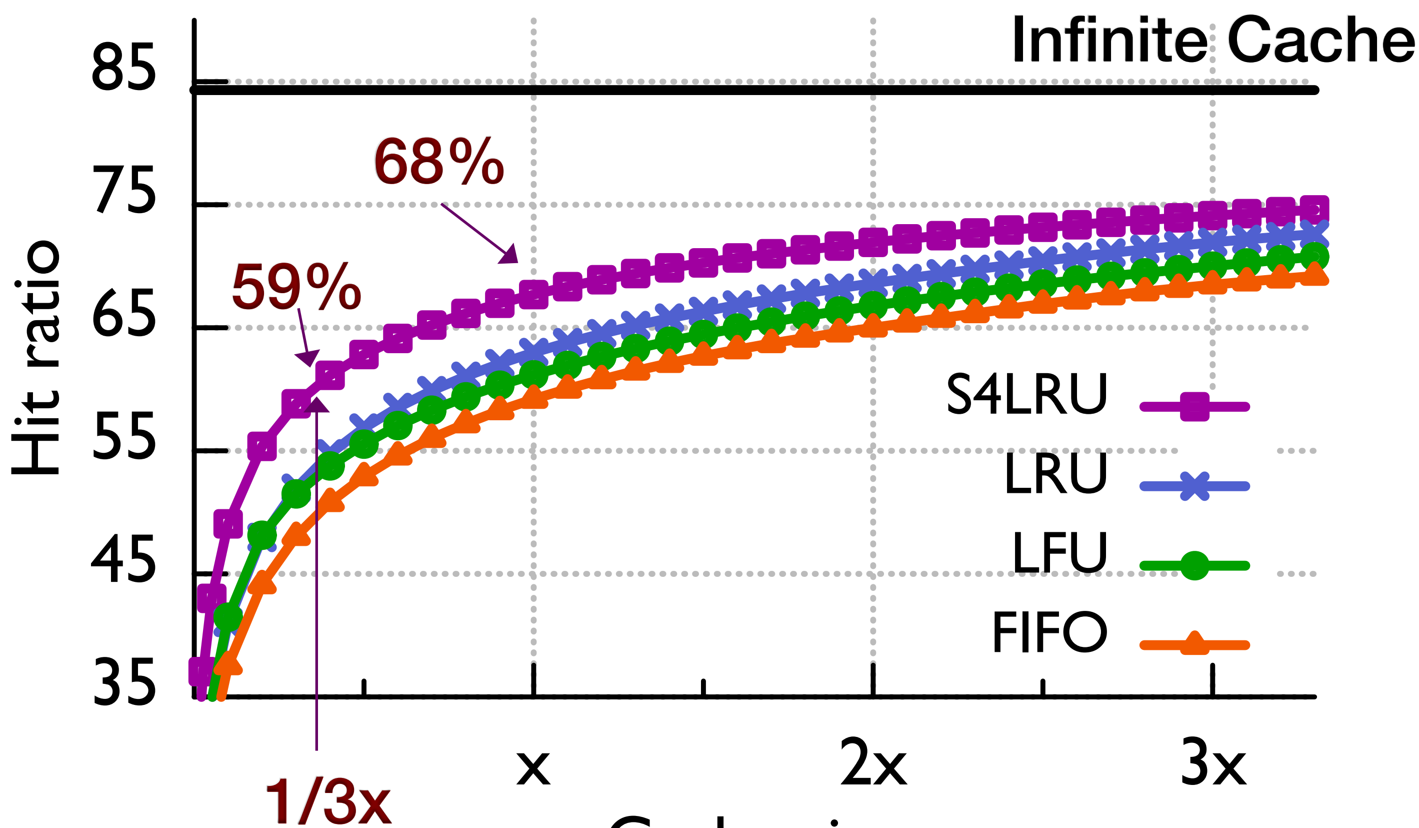
- "Infinite" size ratio needs 45x of current capacity

Edge Cache with Different Algos



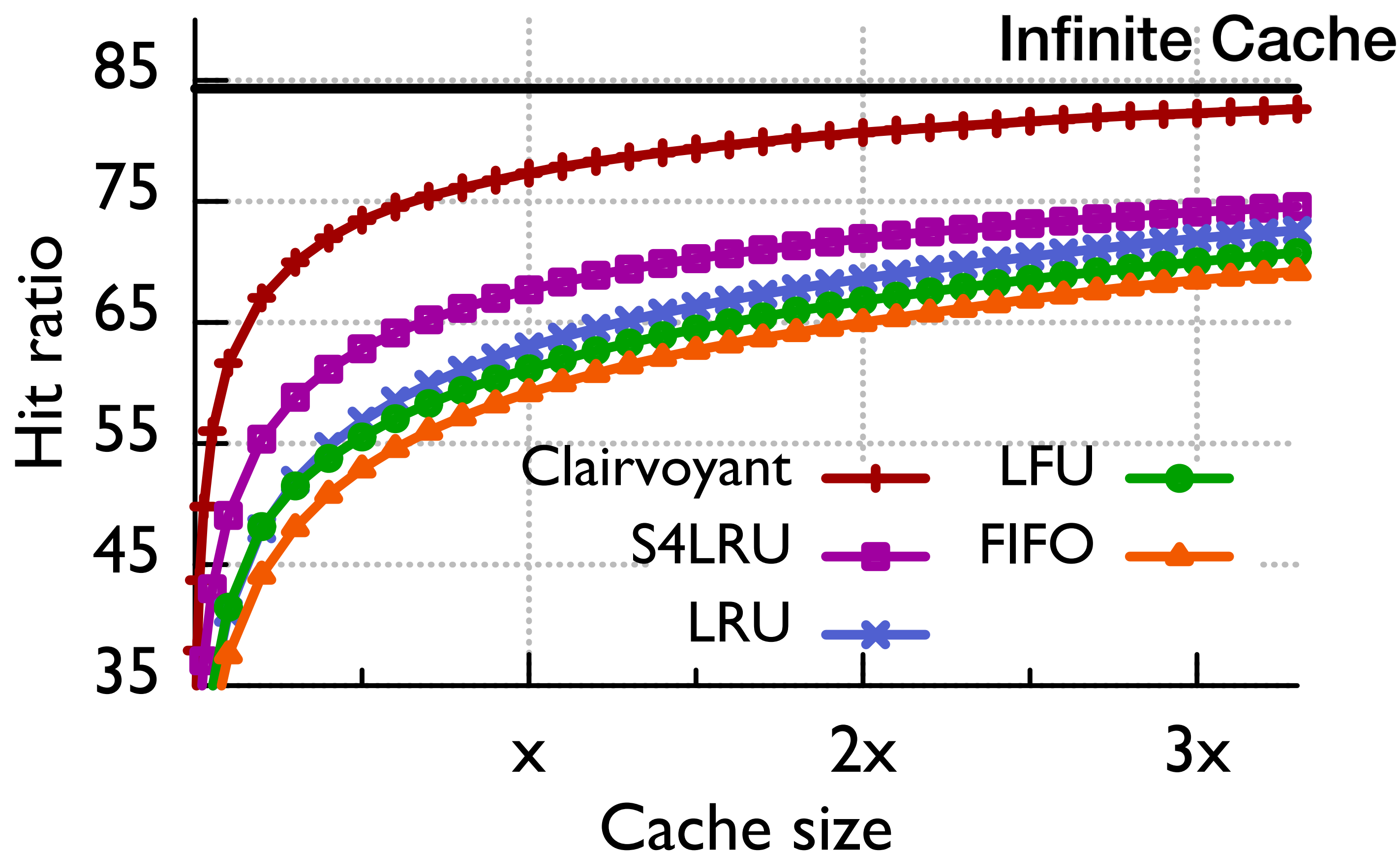
- Both LRU and LFU outperforms FIFO slightly

Edge Cache with Different Algos



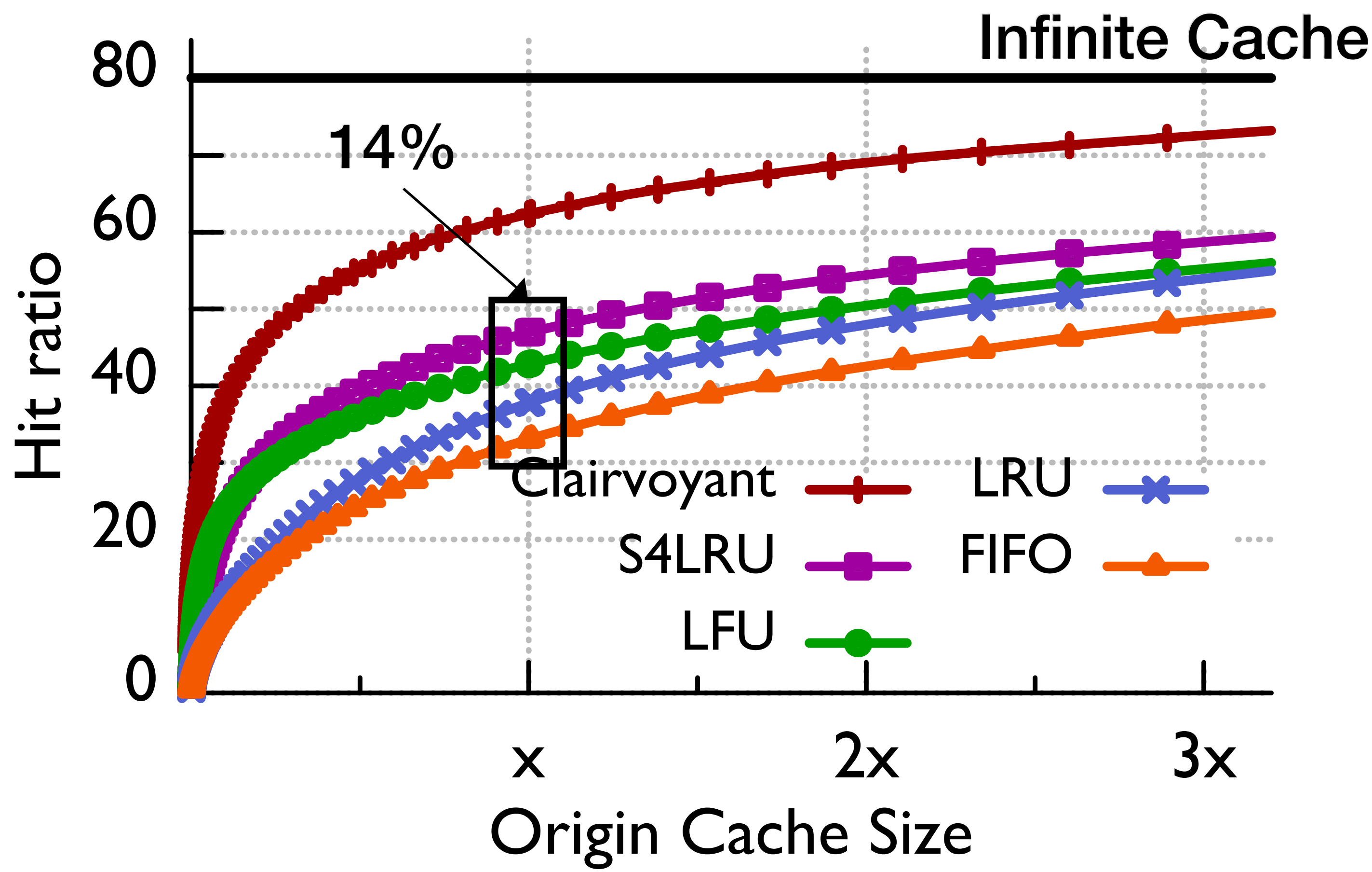
- S4LRU improves the most

Edge Cache with Different Algos



- Clairvoyant (Bélády) shows much improvement space

Origin Cache



- S4LRU improves Origin more than Edge

Content Distribution Networks

- Serve “objects” in web pages, and much more, e.g., video segments
- Load Balancing: Which CDN location?
  - Several options, all of which are interesting distributed systems designed to work with internet routing
- Caching Algorithms: What do we store on them?
  - FIFO, LRU, LFU, … active area of research
- Facebook’s CDN
  - Hierarchy, effectiveness, improvements