Web Caching and Content Delivery

Performance is a major concern in the Web
Proxy caching is the most widely used method to improve Web performance
- Duplicate requests to the same document served from cache
- Hits reduce latency, bandwidth demand, server load
- Misses increase latency (extra hops)

Clients
Proxy Cache
Servers
[Source: Geoff Voelker]

Proxy Caching
How should we build caching systems for the Web?
- Seminal paper [Chankhunthod96]
- Proxy caches [Duska97]
- Akamai DNS interposition [Karger99]
- Cooperative caching [Tewari99, Fan98, Wolman99]
- Popularity distributions [Breslau99]
- Proxy filtering and transcoding [Fox et al]
- Consistency [Tewari, Cao et al]
- Replica placement for CDNs [et al]

Proxy Cache Effectiveness
How to measure Web cache effectiveness (goals)?
- Hit ratio
- Savings in bandwidth or server load
- Reduction in perceived user latency

What factors determine/limit effectiveness?
- Capacity?
- User population?
- Proxy placement in the network?
- Updates and invalidations?

End-to-End Content Delivery

Proxy Cache Effectiveness
How to measure Web cache effectiveness (goals)?
- Hit ratio
- Savings in bandwidth or server load
- Reduction in perceived user latency

What factors determine/limit effectiveness?
- Capacity?
- User population?
- Proxy placement in the network?
- Updates and invalidations?
Web Traffic Characterization

Research question: how do goals and traffic behavior shape strategies for deploying and managing proxy caches?
  - Replacement policy: what objects to retain in cache?
    - Large vs. small, relative importance of popularity and stability
  - Deployment: where to place the cache?
    - Close to server or client?
  - How many users per cache?
  - Prefetching?
Since the Web is in active deployment on a large scale, Web traffic characterization is an empirical science.
  - Science of mass behavior: observe and test hypotheses.

Zipf

[Breslau/Cao99] and others observed that Web accesses can be modeled using Zipf-like probability distributions.
  - Rank objects by popularity: lower rank \( i \) \( \Rightarrow \) more popular.
  - The probability that any given reference is to the \( i \)th most popular object is \( p_i \).
    - Not to be confused with \( p_o \), the percentage of cacheable objects.
Zipf says: \( p_i \) is proportional to \( 1/i^\alpha \), for some \( \alpha \) with \( 0 < \alpha < 1 \).
  - Higher \( \alpha \) gives more skew: popular objects are way popular.
  - Lower \( \alpha \) gives a more heavy-tailed distribution.
  - In the Web, \( \alpha \) ranges from 0.6 to 0.8 [Breslau/Cao99].
  - With \( \alpha = 0.8 \), 0.3% of the objects get 40% of requests.

Zipf-like Reference Distributions

Probability of access to the object with popularity rank \( i \):
\[
p_i = \frac{1}{i^\alpha}
\]

such that:
\[
\sum p_i = 1
\]

(head: flattened, tail: intact)

Importance of Traffic Models

Analytical models like this help us to predict cache hit ratios (object hit ratio or byte hit ratio).
  - E.g., get object hit ratio as a function of size by integrating under segments of the Zipf curve.
  - Assuming perfect LFU replacement
  - Must consider update rate
    - Do object update rates correlate with popularity?
  - Must consider object size
    - How does size correlate with popularity?
  - Must consider proxy cache population
    - What is the probability of object sharing?
  - Enables construction of synthetic load generators
    - SURGE [Barford and Crovella 99]

The “Trickle-Down Effect”

What is the effect on “downstream” traffic?
What is the significance of this effect?
How does it impact design choices for components “behind” the caches?
**What’s Happening? (LRU)**

Suppose the cache fills up in \( R \) references.

(That’s a property of the trace and the cache size.)

Then a cache miss on object with rank \( i \) occurs only if \( i \) is referenced….

- probability \( p_i \)
- …and \( i \) has not been referenced in the last \( R \) requests.

probability \( (1 - p_i)^R \)

*Stack distance*

\[ P(\text{a miss is to object } i) = p_i (1 - p_i)^R \]

**Limitations/Features of This Study**

- static (cacheable) objects
- ignore misses caused by updates
  - invalidation/expiration
- LRU replacement
- vary cache effectiveness by *capacity*
  - cache intercepts all client traffic
- ignore effect on downstream traffic *volume*
Proxy Deployment and Use

Where to put it?
How to direct user Web traffic through the proxy?
Request redirection
  • Much more to come on this topic…
Must the server consent?
  • Protected content
  • Client identity
  “Transparent” caching and the end-to-end principle
  • Must the client consent?

Interception Switches

ISP cache array
The client doesn’t know.
The server doesn’t know.
Neither side told HTTP to disable it.
Is it legal? Good thing? Bad thing?

Shouldn’t This Be Illegal?

end
middle
end

RFC 1122: The Internet Architecture (IPv4) specifies that each packet has a unique destination “host” address.

Problems
middle boxes may be subversive
IPsec and SSL
dynamic routing

Cache Effectiveness

Previous work has shown that hit rate increases with population size
  [Duska et al. 97, Breslau et al. 98]
However, single proxy caches have practical limits
  • Load, network topology, organizational constraints
One technique to scale the client population is to have proxy caches cooperate