Web Caching and CDNs

Aditya Akella
Where can bottlenecks occur?

- First mile: client to its ISPs
- Last mile: server to its ISP
- Server: compute/memory limitations
- ISP interconnections/peerings: congestion inside the network

- Caching at various locations to overcome these bottlenecks
Proxy Caches

Diagram showing the flow of HTTP requests and responses between clients, proxy servers, and origin servers. The diagram illustrates how proxy caches minimize latency and reduce bandwidth usage by caching content locally.
Forward Proxy

• Cache “close” to the client
  – Under administrative control of client-side AS

• Explicit proxy
  – Requires configuring browser

• Implicit proxy
  – Service provider deploys an “on path” proxy
  – ... that intercepts and handles Web requests
Reverse Proxy

• Cache “close” to server
  – Either by proxy run by server or in third-party content distribution network (CDN)

• Directing clients to the proxy
  – Map the site name to the IP address of the proxy
Google Design

Requests

Internet

Private Backbone

Data Centers

Servers

Router

Reverse Proxy

Client

Servers

Router
Limitations of Web Caching

• Much content is not cacheable
  – Dynamic data: stock prices, scores, web cams
  – CGI scripts: results depend on parameters
  – Cookies: results may depend on passed data
  – SSL: encrypted data is not cacheable
  – Analytics: owner wants to measure hits

• Stale data
  – Or, overhead of refreshing the cached data
Proxy Caches

(A) Forward  (B) Reverse  (C) Both  (D) Neither

- Reactively replicates popular content
- Reduces origin server costs
- Reduces client ISP costs
- Intelligent load balancing between origin servers
- Offload form submissions (POSTs) and user auth
- Content reassembly or transcoding on behalf of origin
- Smaller round-trip times to clients
- Maintain persistent connections to avoid TCP setup delay (handshake, slow start)
Content Distribution Networks
Content Distribution Network

• Proactive content replication
  – Content provider (e.g., CNN) contracts with a CDN

• CDN replicates the content
  – On many servers spread throughout the Internet

• Updating the replicas
  – Updates pushed to replicas when the content changes
Server Selection Policy

• Live server
  – For availability

• Lowest load
  – To balance load across the servers

• Closest
  – Nearest geographically, or in round-trip time

• Best performance
  – Throughput, latency, ...

• Cheapest bandwidth, electricity, ...

Requires continuous monitoring of liveness, load, and performance
Server Selection Mechanism

• Application
  – HTTP redirection

• Advantages
  – Fine-grain control
  – Selection based on client IP address

• Disadvantages
  – Extra round-trips for TCP connection to server
  – Overhead on the server
Server Selection Mechanism

• Routing
  – Anycast routing

• Advantages
  – No extra round trips
  – Route to nearby server

• Disadvantages
  – Does not consider network or server load
  – Different packets may go to different servers
  – Used only for simple request-response apps

1.2.3.0/24

1.2.3.0/24
Server Selection Mechanism

- **Naming**
  - DNS-based server selection

- **Advantages**
  - Avoid TCP set-up delay
  - DNS caching reduces overhead
  - Relatively fine control

- **Disadvantage**
  - Based on IP address of local DNS server
  - “Hidden load” effect
  - DNS TTL limits adaptation
How Akamai Works
Akamai Statistics

• Distributed servers
  – Servers: ~100,000
  – Networks: ~1,000
  – Countries: ~70

• Many customers
  – Apple, BBC, FOX, GM
    IBM, MTV, NASA, NBC, NFL, NPR, Puma, Red Bull, Rutgers, SAP, ...

• Client requests
  – Hundreds of billions per day
  – Half in the top 45 networks
  – 15-20% of all Web traffic worldwide
How Akamai Uses DNS

cnn.com (content provider)  DNS root server

End user

GET
index.html

http://cache.cnn.com/foo.jpg

1 2

HTTP

Akamai global
DNS server

Akamai regional
DNS server

Nearby
Akamai
cluster

Akamai
cluster
How Akamai Uses DNS

cnn.com (content provider)  DNS TLD server

DNS lookup
cache.cnn.com

End user

1 2 3 4

ALIAS:
g.akamai.net

Akamai global DNS server

Akamai regional DNS server

Nearby Akamai cluster

Akamai cluster

HTTP
How Akamai Uses DNS

1. **cnn.com (content provider)**
2. **DNS TLD server**
3. **DNS lookup**
   - g.akamai.net
4. **ALIAS**
   - a73.g.akamai.net
5. **Akamai global DNS server**
6. **Akamai regional DNS server**

End user
How Akamai Uses DNS

cnn.com (content provider)  DNS TLD server

End user
How Akamai Uses DNS

cnn.com (content provider) → DNS TLD server

HTTP Request: GET /foo.jpg
Host: cache.cnn.com

End user

Akamai global DNS server → Akamai cluster

Akamai regional DNS server → Akamai cluster
How Akamai Uses DNS

cnn.com (content provider)  DNS TLD server

GET foo.jpg

1 2 3 4 5 6 7 8 9 10 11 12

End user

GET /foo.jpg
Host: cache.cnn.com

Akamai global DNS server

Akamai regional DNS server

Nearby Akamai cluster

Akamai cluster
How Akamai Works: Cache Hit

1. cnn.com (content provider) queries the DNS TLD server.
2. The DNS TLD server queries the Akamai global DNS server.
3. The Akamai global DNS server queries the Akamai regional DNS server.
4. The Akamai regional DNS server queries a nearby Akamai cluster.
5. The nearby Akamai cluster sends the response to the regionally cached DNS server.
6. The DNS regionally cached server sends the response to the end user.
Mapping System

• Equivalence classes of IP addresses
  – IP addresses experiencing similar performance
  – Quantify how well they connect to each other

• Collect and combine measurements
  – Ping, traceroute, BGP routes, server logs
    • E.g., over 100 TB of logs per days
  – Network latency, loss, and connectivity
Mapping System

• Map each IP class to a preferred server cluster
  – Based on performance, cluster health, etc.
  – Updated roughly every minute

• Map client request to a server in the cluster
  – Load balancer selects a specific server
  – E.g., to maximize the cache hit rate