ATTRIBUTION

• These slides are released under an Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0) Creative Commons license
• These slides incorporate material from:
  • Kyle Jamieson, Princeton University (also under a CC BY-NC-SA 3.0 Creative Commons license)
ANNOUNCEMENTS

Homeworks 5 and 6 due this week
Project 2 due Friday at 5pm
OUTLINE

1. Web caching
2. Content-distribution networks
   • Featuring Akamai
   • Overlay networks
WEB CACHING

- Many clients transfer the **same information**
- Generates **redundant** server and network load
- Also, clients may experience high **latency**

![Diagram of web caching network with ISP-1, ISP-2, Backbone ISP, Origin server, and Clients]
WHY WEB CACHING?

Motivation for placing content closer to client:
- User gets better response time
  - Content providers get happier users
- Network gets reduced load

Why does caching work? Exploits locality of reference

How well does caching work?
- Very well, up to a limit
- Large overlap in content
- But many unique requests
CACHING WITH REVERSE PROXIES

- Cache data close to origin server → decrease server load
  - Typically done by content providers
  - Client thinks it is talking to the origin server (the server with content)
- Does not work for **dynamic content**
CACHING WITH FORWARD PROXIES

- Cache close to clients $\rightarrow$ less network traffic, less latency
  - Typically done by ISPs or corporate LANs
  - **Client configured** to send HTTP requests to forward proxy
- Reduces traffic on ISP-1’s access link, origin server, and backbone ISP
CACHING & LOAD-BALANCING: OUTSTANDING PROBLEMS

• Problem *ca. 2002*: How to reliably deliver large amounts of content to users worldwide?

• Popular event: “Flash crowds” overwhelm (replicated) web server, access link, or back-end database infrastructure

• More rich content: audio, video, photos

• Web caching: Diversity causes low cache hit rates (25–40%)
OUTLINE

1. Web caching

2. Content-distribution networks
   • Featuring Akamai
   • Overlay networks
CONTENT DISTRIBUTION NETWORKS

- **Proactive content replication**
  - Content provider (e.g. CNN) pushes content out from its own *origin server*
- **CDN replicates** the content
- On many servers spread throughout the Internet
- **Updating the replicas**
- Updates **pushed to replicas** when the content changes
REPLICA SELECTION: GOALS

- **Live** server
  - For availability

- **Lowest load**
  - To balance load across the servers

- **Closest**
  - Nearest geographically, or in round-trip time

- **Best performance**
  - Throughput, latency, reliability...
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
  - 20+M per second
  - Half in the top 45 networks
  - 20% of all Web traffic worldwide
HOW AKAMAI USES DNS

1. cnn.com (content provider) sends a GET request for index.html to the DNS TLD server.
2. The DNS TLD server resolves the domain to a nearby Akamai cluster.
3. The Akamai regional DNS server sends a response to the End user.
4. The End user's browser requests cache.cnn.com/foo.jpg via HTTP.
5. The Akamai global DNS server sends the response to the End user's browser.
HOW AKAMAI USES DNS

1. End user
2. DNS lookup: cache.cnn.com
3. DNS TLD server
4. ALIAS: g.akamai.net
HOW AKAMAI USES DNS

cnn.com (content provider) -> DNS TLD server

1. DNS lookup for g.akamai.net
2. Akamai global DNS server
3. Nearby Akamai cluster
4. DNS alias a73.g.akamai.net
5. Akamai regional DNS server
6. End user
HOW AKAMAI USES DNS

cnn.com (content provider) -> DNS TLD server

1. Akamai global DNS server
2. Akamai regional DNS server
3. Nearby Akamai cluster
4. Akamai cluster
5. DNS a73.g.akamai.net
6. Address 1.2.3.4
7. End user

HTTP request
HOW AKAMAI USES DNS

1. End user sends a DNS query for "cache.cnn.com".
2. The query is sent to the DNS TLD server.
3. The TLD server forwards the query to the Akamai global DNS server.
4. The global DNS server forwards the query to the Akamai regional DNS server.
5. The regional DNS server forwards the query to the nearest Akamai cluster.
6. The cluster responds with the IP address of the requested content provider.
7. The end user's browser uses the IP address to request the content (GET /foo.jpg Host: cache.cnn.com).
8. The content is served from the Akamai cluster.
9. The content is delivered to the end user.
HOW AKAMAI USES DNS

cnn.com (content provider)  

DNS TLD server

Akamai global DNS server

Akamai regional DNS server

Nearby Akamai cluster

End user

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

GET foo.jpg

1 2 3 4 5 6 7 8 9 10 11 12
HOW AKAMAI USES DNS

cnn.com (content provider) ----> DNS TLD server

End user ----> Nearby Akamai cluster

Akamai global DNS server

Akamai regional DNS server

Akamai cluster
HOW AKAMAI WORKS: CACHE HIT

cnn.com (content provider) → DNS TLD server

1. HTTP request (cnn.com)
2. DNS query (cnn.com)
3. DNS response (Akamai global DNS server)
4. DNS response (Akamai regional DNS server)
5. TCP connection to nearby Akamai cluster
6. Content delivery to 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, throughput, and connectivity
Map each **IP class** to a preferred **server cluster**

- Based on performance, cluster health, etc.
- Updated roughly every minute
  - **Short, 60-sec DNS TTLs** in Akamai regional DNS accomplish this

Map client request to a server in the cluster

- **Load balancer** selects a specific server
- *e.g.*, to **maximize** the **cache hit rate**
ADAPTING TO FAILURES

- Failing **hard drive** on a server
  - Suspends after finishing “in progress” requests

- Failed **server**
  - Another server takes over for the IP address
  - Low-level map updated **quickly** (load balancer)

- Failed **cluster**, or **network path**
  - High-level map updated **quickly** (ping/traceroute)
TAKE-AWAY POINTS: CDNS

• Content distribution is hard
  • Many, diverse, changing objects
  • Clients distributed all over the world

• Moving content to the client is key
  • Reduces latency, improves throughput, reliability

• Content distribution solutions evolved:
  • Load balancing, reactive caching, to
  • Proactive content distribution networks
OUTLINE

1. Web caching
2. Content-distribution networks
   • Featuring Akamai
   • Overlay networks
OVERLAYS FOR ROUTING: WHY?

- Triangle inequality doesn’t hold in networks!
OVERLAY NETWORKS FOR ROUTING

- Underlying network
- Internet connectivity (IP Routing)
• Potential overlay connectivity
• SF as root
OVERLAY NETWORKS

- Determine edge weights
- E.g., bandwidth, latency
• Build overlay connectivity
• An application-layer distribution tree