Content Distribution Networks (CDNs)

Mike Freedman
COS 461: Computer Networks
Lectures: MW 10-10:50am in Architecture N101

http://www.cs.princeton.edu/courses/archive/spr13/cos461/

Second Half of the Course

- Application case studies
  - Content distribution, peer-to-peer systems and distributed hash tables (DHTs), and overlay networks

- Network case studies
  - Enterprise, wireless, cellular, datacenter, and backbone networks; software-defined networking

- Network security
  - Securing communication protocols
  - Interdomain routing security

Single Server, Poor Performance

- Single server
  - Single point of failure
  - Easily overloaded
  - Far from most clients

- Popular content
  - Popular site
  - “Flash crowd” (aka “Slashdot effect”)}
  - Denial of Service attack

Skewed Popularity of Web Traffic

“Zipf” or “power-law” distribution

Characteristics of WWW Client-based Traces
Carlos R. Cunha, Azer Bestavros, Mark E. Crovella, BU-CS-95-01
Web Caching

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
**Proxy Caches**

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

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

**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
Modern HTTP Video-on-Demand

- Download “content manifest” from origin server
- List of video segments belonging to video
  - Each segment 1-2 seconds in length
  - Client can know time offset associated with each
  - Standard naming for different video resolutions and formats:
    e.g., 320dpi, 720dpi, 1040dpi, ...
- Client downloads video segment (at certain resolution) using standard HTTP request.
  - HTTP request can be satisfied by cache: it’s a static object
- Client observes download time vs. segment duration, increases/decreases resolution if appropriate

Content Distribution Networks

- 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

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

1. DNS lookup
2. cache.cnn.com
3. g.akamai.net
4. ALIAS: g.akamai.net
5. Nearby Akamai cluster
6. 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

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

- Failed cluster
  - High-level map updated quickly

- Failed path to customer’s origin server
  - Route packets through an intermediate node
Akamai Transport Optimizations

- Bad Internet routes
  - Overlay routing through an intermediate server
- Packet loss
  - Sending redundant data over multiple paths
- TCP connection set-up/teardown
  - Pools of persistent connections
- TCP congestion window and round-trip time
  - Estimates based on network latency measurements

Akamai Application Optimizations

- Slow download of embedded objects
  - Prefetch when HTML page is requested
- Large objects
  - Content compression
- Slow applications
  - Moving applications to edge servers
  - E.g., content aggregation and transformation
  - E.g., static databases (e.g., product catalogs)

Conclusion

- Content distribution is hard
  - Many, diverse, changing objects
  - Clients distributed all over the world
  - Reducing latency is king
- Contribution distribution solutions
  - Reactive caching
  - Proactive content distribution networks