Content Delivery Networks

Operating Systems – COSC 6360

Purvi Shah

purvi@cs.uh.edu
What Is In This Presentation?

• Here look at a larger-scale problem typical of the modern content delivery networks (CDNs)
• Topic relates to how data is served up by high-volume web sites
• What CDN techniques are being used?
• No standard, well supported, solution!
Motivation

• What is a CDN?
  – A overlay network infrastructure delivering content on behalf of an origin site

• State of CDNs
  – A number of CDN companies
    • E.g. Akamai
    • Do content delivery using all sorts of proprietary tricks
    • Mechanisms aren’t standard and are hard to implement
  – Used by many popular origin sites
    • E.g., CNN, CNBC, …
Service model

- Client-Server model
  - request-reply
Client-Server model

- Possible issues
  - scalability
A problem...

- Feb 3: Google linked banner to “julia fractals”
- Users clicking directed to Australian University web site
- …University’s network link overloaded, web server taken down temporarily…
The problem strikes again!

- Feb 4: Slashdot ran the story about Google
- …Site taken down temporarily…again
Service model: proxy

- Client-Intermediary-Server: client approach
Service model: content delivery

- **Client-Intermediary-Server:server approach**
Content distribution networks

- Client-Intermediary-Server model
  - intermediary: content distribution servers
- Origin server and CDN server
  - how to distribute content properly
  - CDN server placement
  - content consistency control
- CDN server and ordinary client
  - how to choose the right CDN server
What CDN redirection techniques are being used?

- Techniques generally used
  - DNS redirection (DR)
  - URL rewriting (UR)
  - HTML redirection
  - Hybrid scheme (URDR)
    - URL rewriting + DNS redirection
- Goal: Transparent to the end-users
- If not ... Use Peer-to-Peer Systems
Akamai content delivery

• Akamai EdgePlatform
  – 15,000+ servers
  – 1,100+ networks
  – 69 countries
  – up to 15% web traffic

• Server selection
  – DNS-based
  – for site or object delivery
This lecture

• Content distribution networks
  – why service models other than “client-server”
  – web caching vs. content delivery
  – DNS-based server selection

• Explore further
  – http://www.akamai.com
Next

Existing CDN approaches

• Client-side proxying
  – Next Class…

• Throw money at the problem
  – Load-balanced servers, fast network connections
  – Problem: Can’t afford or don’t anticipate need

• Content Distribution Networks (CDNs)
  – Akamai
  – Centrally managed, needs to recoup costs
Coral’s solution…

Pool resources to dissipate flash crowds

- Implement an open CDN
- Works with unmodified clients
- CDN only fetches once from origin server
Coral’s solution…

Pool resources to dissipate flash crowds

- Strong locality without a priori knowledge
- No hotspots in CDN
- Should all work automatically with nobody in charge
Using Coral CDN

• Rewrite URLs into “Coralized” URLs


  – Directs clients to Coral, which absorbs load

• Who might “Coralize” URLs?

  – Web server operators Coralize URLs
  – Coralized URLs posted to portals, mailing lists
  – Users explicitly Coralize URLs
CoralCDN components

DNS Redirection
Return proxy, preferably one near client

Origin Server

httpprx

CoralCDN components

httpprx

Resolver

Browser

Fetch data from nearby

Cooperative Web Caching

www.x.com.nyud.net
Functionality needed

• **DNS**: Given network location of resolver, return a proxy near the client

  \[
  \text{put (network info, self)}
  \]

  \[
  \text{get (resolver info)} \rightarrow \{\text{proxies}\}
  \]

• **HTTP**: Given URL, find proxy caching object, preferably one nearby

  \[
  \text{put (URL, self)}
  \]

  \[
  \text{get (URL)} \rightarrow \{\text{proxies}\} \]
Contributions

• Self-organizing clusters of nodes
  – NYU and Columbia prefer one another to Germany

• Rate-limiting mechanism
  – Everybody caching and fetching same URL does not overload any node in system

• Decentralized DNS Redirection
  – Works with unmodified clients

No centralized management or *a priori* knowledge of proxies’ locations or network configurations
CoralCDN components

DNS Redirection
Return proxy, preferably one near client

Cooperative Web Caching

Fetch data from nearby

Resolver

Browser

www.x.com.nyud.net

httpprx

dnssrv
Coral CDN components

DNS Redirection
Return proxy, preferably one near client

www.x.com.nyud.net

Fetch data from nearby
Cooperative Web Caching
Coral’s DNS Redirection

- Coral DNS server probes resolver
- Once local, stay local
  
  When serving requests from nearby DNS resolver
  
  – Respond with nearby Coral proxies
  – Respond with nearby Coral DNS servers
    
    → Ensures future requests remain local

- Else, help resolver find local Coral DNS server
DNS Measurement Mechanism

• Return servers within appropriate cluster
  – e.g., for resolver RTT = 19 ms, return from cluster < 20 ms

• Use network hints to find nearby servers
  – i.e., client and server on same subnet

• Otherwise, take random walk within cluster
Experimental results

• Consider requests to Australian web site:
  – Does Coral absorb flash crowds?
  – Does clustering help latency?
  – Does Coral form sensible clusters?
  – Does Coral prevent hotspots?

• Experimental setup
  – 166 PlanetLab hosts; Coral node and client on each
  – Twelve 41-KB files on 384 Kb/sec (DSL) web server
  – (0.6 reqs / sec) / client → 32,800 Kb/sec aggregate
Solves flash-crowd problem

Coral hits in 20 ms cluster

Local caches begin to handle most requests

Hits to origin web server
Benefits end-to-end client latency
Benefits end-to-end client latency
Conclusions

• Coral indexing infrastructure
  – Uses non-standard P2P
  – Stores network hints and forms clusters
    • Exposes hierarchy and hints to applications
  – Prevents hotspots

• Use Coral to build fully decentralized CDN
  – Solves Slashdot effect
  – Popular data → widely replicated → highly available
    • Democratizes content publication
Next Class

Contact: purvi@cs.uh.edu

Thanks!