CDN Overview and Discussion for IRIS-HEP

John A. Dilley

Introduction

Summary of terms used in this discussion

- Content: internet information (HTML pages, images, videos, downloads)
- Delivery: transmission of content from a server to an endpoint (user/browser, device)
- CDN: a service that delivers content on behalf of a content owner
- HTTP/TCP/IP: connection-oriented transmission protocol, lingua franca of the web
- Request: a single HTTP operation; mostly GET (fetch an object), some POST or PUT

- Latency: endpoint-to-server per packet round-trip-time (RTT) across the IP network
- Response Time: click-to-done time for a web page, often comprising many requests

CDN Benefits

Simplify resource provisioning for content owner

Accelerate delivery through low RTT, scalable delivery architecture

- Low RTT speeds connection setup (TCP/IP), and secure session if using TLS
- Availability via having resources in many locations helps absorb traffic spikes
- Direction of endpoint requests to nearby *available* resources
- Greater elasticity of server resource than typical single origin server

CDN Operational Model

Machines are operated by CDN provider, outsourcing capex and operations

- CDN owner establishes many relationships with data center providers
- CDN owner handles network and system management (24x7 operations)
- Deploy and monitor servers, keeping ahead of aggregate scaling demands
- Keep servers updated (security patches) and operating efficiently

CDN Deployment Modes

Akamai model: deploy in many locations in an attempt to get servers close to many/most users

- Optimal cluster size in this model varies, often 4-20 machines in a given location
- Machines typically are disk heavy, 6-8 disks not unusual; SSD as well for performance
- CPU in a well tuned CDN server is mostly a "bit shovel" for cacheable workloads

Other CDN/delivery models use fewer, larger locations

- Being a few more network hops to larger installations eases operational burden (scaling)
- Also easier to operate in fewer large ISPs than thousands of large-to-small ISPs (relationships)

Tradeoffs: lower latency (higher end user performance) for higher operational complexity

Cacheable Content Delivery

CDN quickly delivers repeated content requests from cache (disk, memory)

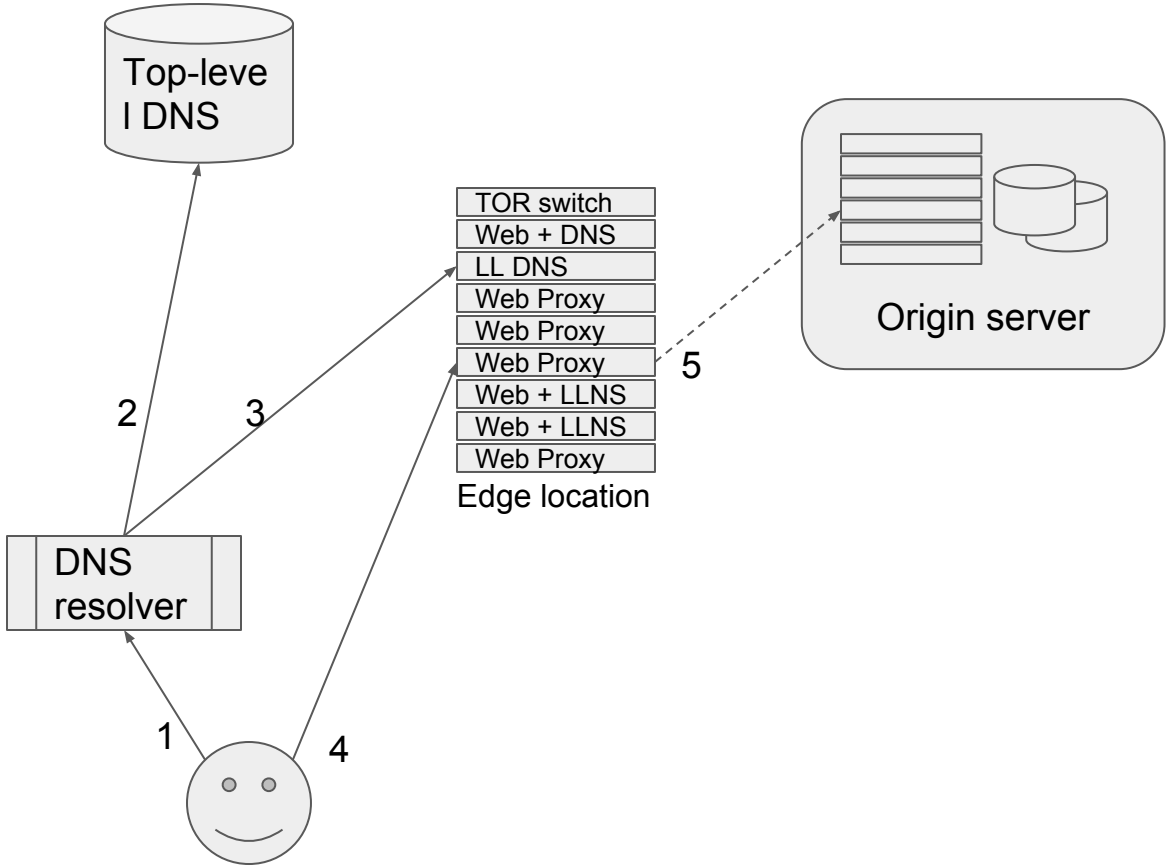
- Second to Nth request often served from edge without WAN interaction
  - Web content has high recurrence -- 90+% cache hit rates common
  - High memory hit rate for modest-sized objects (not so much for downloads/large data sets)
- Generally use a demand-driven cache model for efficiency
- Time-to-live based cache coherence triggers refresh check after TTL expires

Challenges

- Uncacheable / whole-site delivery, operations at scale, mapping and steering
- Tuning for high performance (manage infrastructure cost at scale)

DNS Based CDN

1. End user requests IP for hostname
2. Resolver asks CDN's TL DNS for IP, gets LL DNS in "best" edge
3. Resolver asks LL DNS for IP, gets 1+ IPs for the requested hostname
4. End user browser sends request to IP of CDN proxy
5. CDN web proxy sends request to "origin" server (if required), caches response object if permitted, sends response to end user



Cache Optimization

- Consistent hashing optimization for placement of objects within clusters
  - Places objects on minimum number of servers (min two) in each location
  - When cluster changes no object movement is required immediately
  - Future requests gradually redistribute content to new/different servers
  - More servers can be brought to service to handle load spikes (spreading)

To support large footprint content store it only once per cluster

- Identify primary, route requests within cluster to primary
- Trades off cluster footprint for availability and performance
  - If primary down: all requested content must be loaded onto the secondary
  - Additional network hop within cluster roughly doubles CPU for cacheable content

WAN Optimization

Large footprint allows optimization of wide area network requests via overlay network

Optimizations at HTTP, TCP, and IP layers

- HTTP: persistent connections (especially to CDN peers), transfer-encoding gzip
- TCP: optimize transport settings from CDN endpoint forward to origin (or peer)
- IP: identify impacted routes (congestion, BGP damage) and route around

CDN-based Security

Security services are a natural fit with CDN

- Prevent inbound attacks -- deflect garbage traffic (UDP, SYN flood)
- Absorb (and deliver) “clean” high volume requests

Scan for content attacks (also should do this at origin server)