Consistency Issues

• Web pages tend to be updated over time
  – Different update frequencies (few minutes to few weeks)
• How can a proxy cache maintain consistency of cached data?
  – Push versus pull
  – Send invalidate or update

Push-Based Approach

• Server tracks all proxies that have requested objects
• If a web page is modified, notify each proxy
• Notification types
  – Indicate object has changed [invalidate]
  – Send new version of object [update]
• How to decide between invalidate and updates?
  – Send updates for popular objects, invalidate for rest
Push-Based Approach

• Advantages
  – Provide tight consistency [minimal stale data]
  – Proxies can be passive
• Disadvantages
  – Need to maintain state at the server
    • Recall that HTTP is stateless
    • Need mechanisms beyond HTTP
  – State may need to be maintained indefinitely
    • Not resilient to server crashes

Pull-Based Approach

• Proxy is entirely responsible for maintaining consistency
• Proxy periodically polls the server to see if object has changed
  – Use If-Modified-Since HTTP message
• Key question: when should a proxy poll?
  – Server assigns an expiration time to page
    • No guarantee if the object will change in the interim
Pull-Based Approach: Intelligent Polling

• Proxy can dynamically determine the refresh interval
  – Start with a conservative refresh interval
  – Increase interval if object has not changed between two successive polls
  – Decrease interval if object is updated between two polls
  – Adaptive: No prior knowledge of object characteristics needed
• Can maintain strict consistency by always sending conditional GET requests.

Pull-Based Approach

• Advantages
  – Implementation using HTTP (If-modified-Since)
  – Server remains stateless
  – Resilient to both server and proxy failures
• Disadvantages
  – Weaker consistency guarantees (objects can change between two polls and proxy will contain stale data until next poll)
    • Strong consistency only if poll before every HTTP response
  – More sophisticated proxies required
  – High message overhead
A Hybrid Approach: Leases

• Lease: duration of time for which server agrees to notify proxy of modification
• Issue lease on first request, send notification until expiry
  – Need to renew lease upon expiry
• Smooth tradeoff between state and messages exchanged
  – Zero duration => polling, Infinite leases => server-push
• Efficiency depends on the lease duration

Policies for Lease Duration

• Age-based lease
  – Lease duration proportional to age of object
  – Age depends on the last modification time
• Renewal-frequency-based
  – Lease duration proportional to popularity of object
• Server-load-based
  – Shorter leases during heavy load
  – Shorter lease ⇒ less server state
The Principle of Cooperative Caching

1. Look in local cache
2. Ask neighboring proxy caches
3. Forward request to Web server

Hierarchical Proxy Caching
Locating Data Using Directories

Properties
- Lookup is local
- Hit at most 2 hops
- Miss at most 2 hops (1 extra on wrong hint)
- Need update directory whenever a proxy caches/discards an object

Content Distribution Networks (CDNs)

- A CDN company installs CDN servers (proxies) throughout the Internet
  - Content provider pushes content to the CDN distribution node
  - CDN distribution node pushes the content to selected CDN servers
- Which proxy answers a client request?
  - Ideally the “closest” proxy
  - Akamai uses a DNS-based approach
Akamai CDN

Using DNS to direct requests to a nearby CDN server

Replication of Web Applications

Alternatives for caching and replication with Web applications