Electronics and Computer Science
University of Southampton

CM214-COMP2008

Data Communications and Networks

Protocol Caching

Karl R. Wilcox
krw@ecs.soton.ac.uk

Caching

- The storing of something in a hidden place for use later
- We will consider in detail web caching
  - Storage of resources transferred over the HTTP protocol
- In particular, the extensions to the protocol to explicitly assist caching

Web Caches

- Why have Web Caches?
  - Reduce Latency
  - Traffic Reduction
- Caching is good for users
  - Faster page load times
  - May reduce network costs
- Caching is good for webmasters
  - Faster sites are used more often
  - Reduces load on the server
  - Small, uncacheable files can be used to track usage

Browser Web Cache

- Browser cache (on user's PC) - assumed to be "personal"
  - Usually checked once per session
  - Can be set to check on each use
  - May additionally cache form data
  - Good for "back" and "forward" navigation
  - Can re-use "off-line"
  - What should "refresh" mean? (redraw or reload?)

Proxy Web Cache

- Extension of a Web Proxy (unit 11)
- Proxy Cache - assumed to be "shared“
  - Stored on web proxy
  - Can achieve 50% hit rate
  - Depends on user population

Web Caching Strategies

- Typically, everything is cached except -
  - Resources with HTTP 1.1 Header
  - cache-control: no-cache
  - (Possibly) resources with HTML META tag
  - pragma=no-cache)
    - (usually only the Browser cache reads the HTML)
  - Authenticated pages
  - Secure (SSL) pages
  - Resources without "validation" information, e.g.
    - Without last-Modified or Expires headers
  - Whatever the cache administrator decides not to cache
    - (depends on the capabilities of the cache)

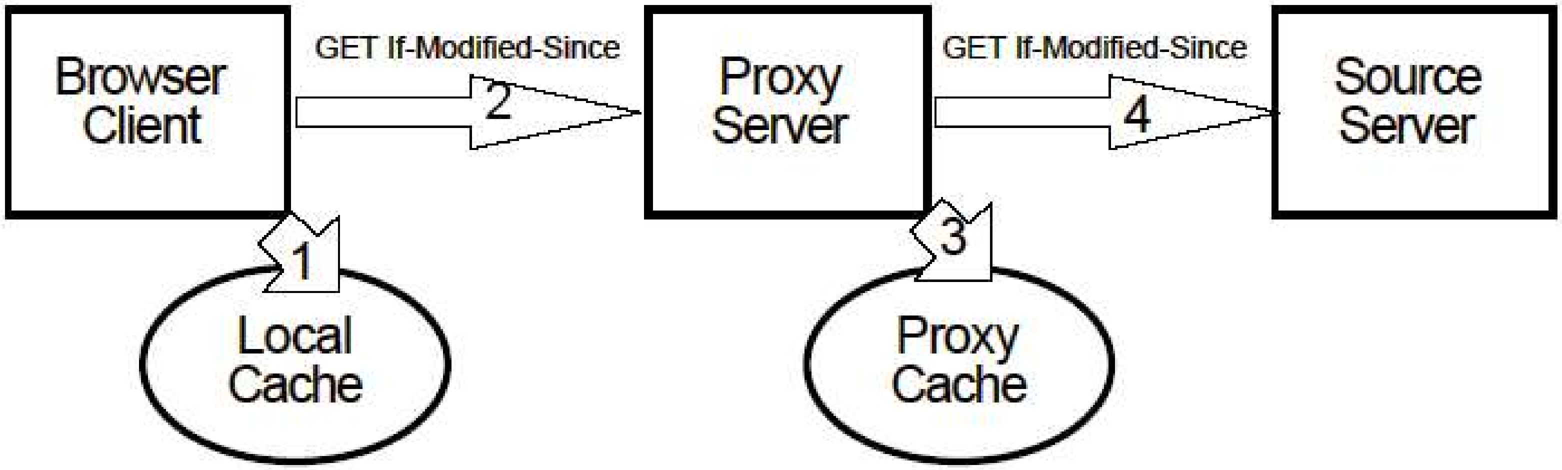
Serving From the Cache

- When is a resource served from the cache?
  - If the resource is present in the cache(!)
  - i.e. it has the same URL ("/img.gif"!= "img.gif")
  - If the "validation" information suggests it is still "fresh"
    - e.g. Still within a given "Expires" time
  - If the resource has already been validated this session
  - (From local cache, if "check once per session" is set)
  - If the Proxy Cache has validated it recently and it was "last-Modifed" a long time ago
    - Note - there is an assumption here!

Resource Validation

- What if the resource is present, but not "fresh"?
  - Validate the resource
  - HEAD (HTTP 1.0)
  - GET If-Modified-Since (HTTP 1.1)
  - E-TAG (HTTP 1.1) - Unique ID for each version of resource
- Note - this validation may come from the proxy, not the source

Validation Example



Validation Headers

- `Expires: Tue, 14 May 2004 18:00:00 GMT`
- `Cache-Control:`
- `max-age=[seconds]` - Like expires, but relative time
- `s-max-age=[seconds]`– As above, but explicitly for proxies
- `public` - Cacheable, even if not normally so
- `no-cache` - Not cacheable, ever
- `must-revalidate` - Enforces use of this information
- `proxy-revalidate` - As above, explicitly for proxies

Caching Issues

- Caches can become out of synchronisation
  - Incorrectly set Expires or max-age
  - Incorrect cache assumptions or clock settings
  - What to do if a resource is stale and the source server is not contactable?
    - Should respond but with a "Warning" header?
  - What about "incorrect" responses (e.g. wrong length)?
- Browsers - SHIFT-Reload issues a no-cache request
- RFC 2616 has an excellent description of caching issues
  - But note how many statements include "MAY"!

Other Cache Types

- FTP
- NFS
- Streaming Media
- Caching is explicitly supported in some protocols
  - HTTP 1.1 - (not HTTP 1.0)
  - Some streaming media
- Caching of some form is generally useful but does add complexity
- Consider it carefully when designing your own protocols!

Summary

- Caching is a generally useful technique for
  - Increasing apparent response time
  - Decreasing network usage
- HTTP 1.1 has explicit support
  - But also issues
- Think carefully before using caching!