HTTP Caching

Prof. Jim Whitehead
CMPS 183 – Spring 2006
May 12, 2006
HTTP Cache Goals

• Resources on the Web are not accessed uniformly.
  – A Zipf distribution (power law distribution)
• Many resources do not change frequently
  – Those that do may be accessed many times in-between changes
• Storing copies of pages closer to the client might result in faster page views, and less network traffic
• HTTP Caches perform this page copying
Concepts in Caching

• Fresh vs. stale
  – When a cached page is the same as the origin, it is fresh
  – When a cached page is different from the origin, it is stale

• Cache expiration
  – When a cache entry goes from being fresh to stale

• Cache validation
  – Determining whether a cached entry is fresh or stale
HTTP Headers for Cache Control

- **Age**
  - *Origin*: How long ago was a response generated on the origin server
  - *Cache*: How long ago was the cached response revalidated

- **Cache-control**
  - Used in both HTTP requests and responses

- **ETag**
  - Unique identifier for entities associated with a URL

- **Vary**
  - Indicates which headers are used to pick resource variants
Cache Control Request Directives

• These are cache control requests made by the client
  – The server has some too… stay tuned
  – Cache-Control: {directive}

• no-cache
  – Forces cache and origin server to return new content. Returning cached copies not allowed
  – Also: Pragma: no-cache

• only-if-cached
  – Client requests only cached responses

• no-store
  – Prevents request and response from being cached
  – Helps meet privacy requirements by ensuring information is not stored in a cache
More Cache Control Request Directives

• max-age
  – Specifies the maximum allowable age of a cached response
  – max-age=0 forces end-to-end revalidation

• max-stale
  – If client is willing to accept slightly stale responses, this indicates how stale they can get before they need to be refreshed.

• min-fresh
  – Client is expressing a desire for the returned response to be fresh for at least as long as specified
  – min-fresh=60 :: response must be fresh for at least a minute

• no-transform
  – Indicates that no modification of the cached response body must be made
Cache Control Response Directives

• public
  – The server is stating that the response can be generally cached

• private
  – The response can be cached, but the cached copy can only be returned to the recipient of the response. Caches cannot return the cached copy to other individuals.

• no-store
  – The response is not allowed to be stored in a cache

• no-cache
  – The response can be stored in a cache, but the cache must validate the cached response before returning it.
  – Provides some efficiency benefit
More Cache Control Response Directives

- **no-transform**
  - Cache must not modify the response when returning it

- **must-revalidate**
  - Forces the cache to revalidate against the origin server before returning cached copy.
  - Similar to no-cache
    - Difference is caches are unlikely to store no-cache responses, but will store must-revalidate ones

- **max-age**
  - Specifies the expiration time of a response
ETag Use

• Server returns ETag with a response
• This is a unique identifier for the entity in the response
• Cache and/or Client stores the ETag
• Can use ETag with If-Match or If-None-Match headers
  – If-Match: if the server’s entity matches the supplied Etag, perform the operation
  – If-None-Match: if the server’s entity does not match the supplied Etag, perform the operation