Cache Digests for HTTP/2

Kazuho Oku
Pull Request #413

• proposes:
  – SENDING_CACHE_DIGEST SETTINGS Parameter
  – switch to Cuckoo hashing
  – thanks to Yoav Weiss for the proposal
• a SETTINGS parameter sent by client
  – “I’m going to send CACHE_DIGEST, so the server should decide what to push after seeing the digest”

• discussion:
  – CACHE_DIGEST frame has “origin” field. Do we need to associate the flag to each origin, or is SETTINGS parameter fine?
  – do we need a way to retract the announcement?
    • i.e. “I said I am going to send CACHE_DIGEST, but I cannot”
Switch to Cuckoo Hashing

- **client-side:**
  - no need to iterate through the browser cache when generating the CACHE_DIGEST frame
  - the hash becomes a persistent structure in the browser cache
    - allowing O(1) insertion *and removal* of URLs
    - O(N) when resizing happens
      - resizing requires additional data to be associated to the entries of the hash
- **server-side:**
  - no need to decode the frame before lookup
- *digest becomes slightly(?) larger*
Cuckoo Hashing – no distinction bet. fresh vs. stale

• what should we include in the digest?
  a) hash(URL):
    • meaningless for stale-cached responses
    • waste of bandwidth if the majority of cached responses are stale
  b) hash(URL+etag):
    • server needs to know the etag of the resource it might push in order to determine if it should push
    • client needs to respect (i.e. save) the pushed response even if it already has a freshly-cached object with the same URL

• pull request proposes b
Cuckoo Hashing – the options

• a) replace Golomb Coded Sets with Cuckoo Hashing?
• b) define both algorithms?
  – might not be an option due to the stale vs. fresh distinction
• c) stick to using Golomb Coded Sets
  – can be generated by browser from Cuckoo Hash with additional data