Browser Caching & You

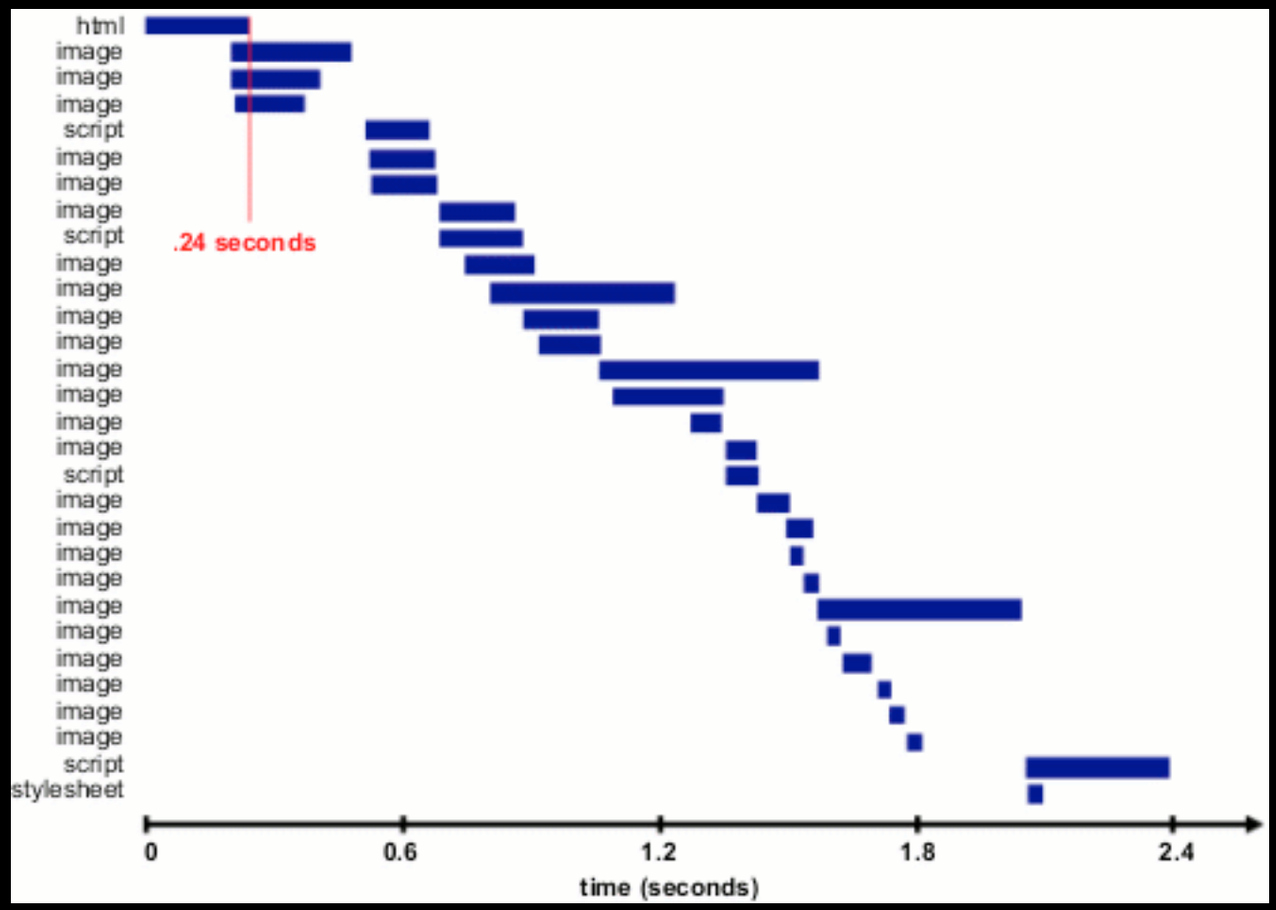
(a love story)

Mark Nottingham / mnot.net

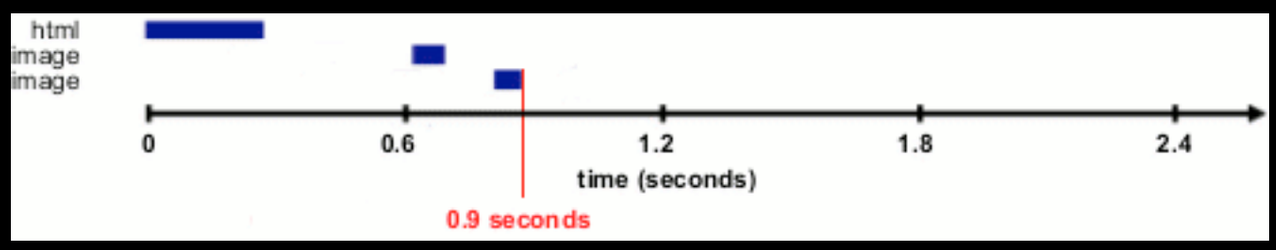
At First Sight:

The Honeymoon

EMPTY

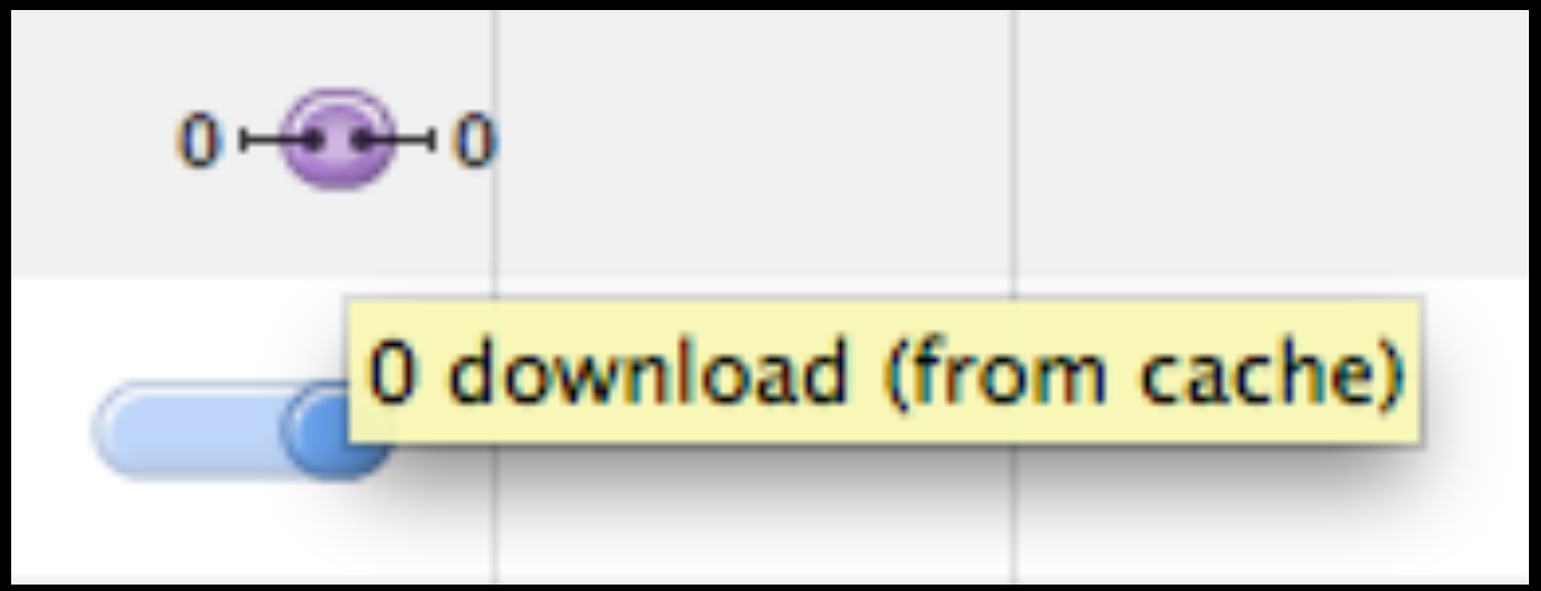


FULL

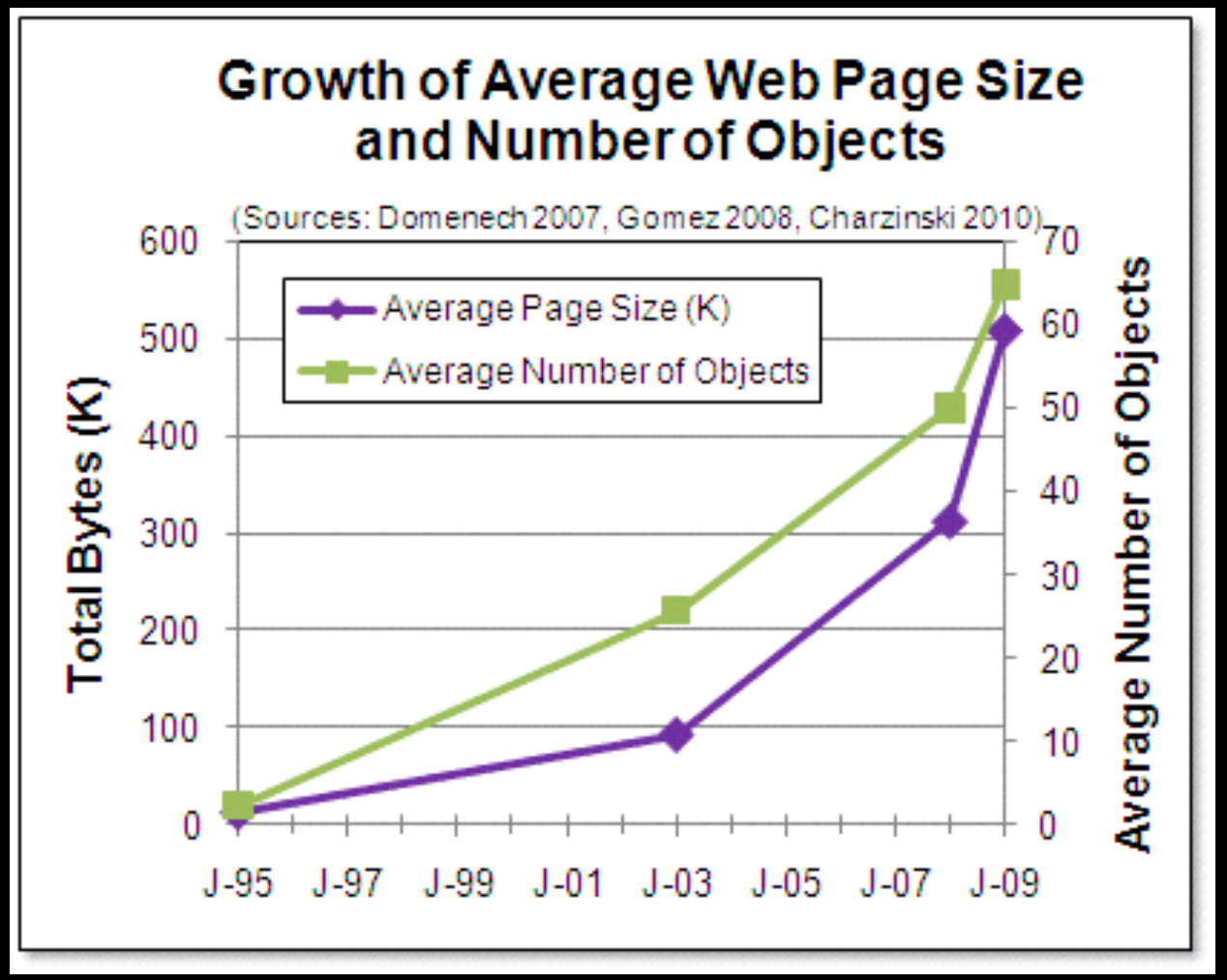


http://www.yuiblog.com/blog/2007/01/04/performance-research-part-2/

Status Code: 304 Not Modified



Relationship Trouble



http://www.websiteoptimization.com/speed/tweak/average-web-page/

Mozilla: 50MB

IE: 10% (max 1G)

Safari: ??

Chrome: 8-250M?

Mozilla: 100 pages

Mozilla: 16384 entries

```
89 // Min and max values for the number of records in the DiskCachemap
90 #define kMinRecordCount    512
91
92 #define kSeparateFile      0
93 // #define must always  be <= 65535KB, or overflow. See bug 443067 Comment 8
94 #define kMaxDataFileSize   5 * 1024 * 1024  // 5 MB (in bytes)
95 #define kBuckets           (1 << 5)    // must be a power of 2!
```

(~230 pages)

```
Cache-Control: max-age=0,
  no-cache, no-store,
  private, pre-check=0,
  post-check=0
```

“Maybe we should start seeing other people...”

HTML5 AppCache

Allow this website to use space on your disk?

The website "https://twitter.com" is requesting 1 MB of disk space to store "html5 test db" as a database on your disk.

?

Don't Allow

Allow

- privacy concerns (“EverCookie”)
- relationship with HTTP caching is poorly defined
- designed for offline use

Getting to know each other

XMLHttpRequest Caching Tests

nearby: X

2008-12-09

This is a set of functional tests for determining how client-side HTTP caches operate.

Note that while they are implemented using XMLHttpRequest (a JavaScript HTTP client), most implementations should use the same cache as for "normal" requests. However, there may be variations.

Each group of tests explains what is being tested and what the implications of failure are. Although many of the tests are automated, some may require user interaction, via a "run test" button. Be sure to follow any *instructions* carefully.

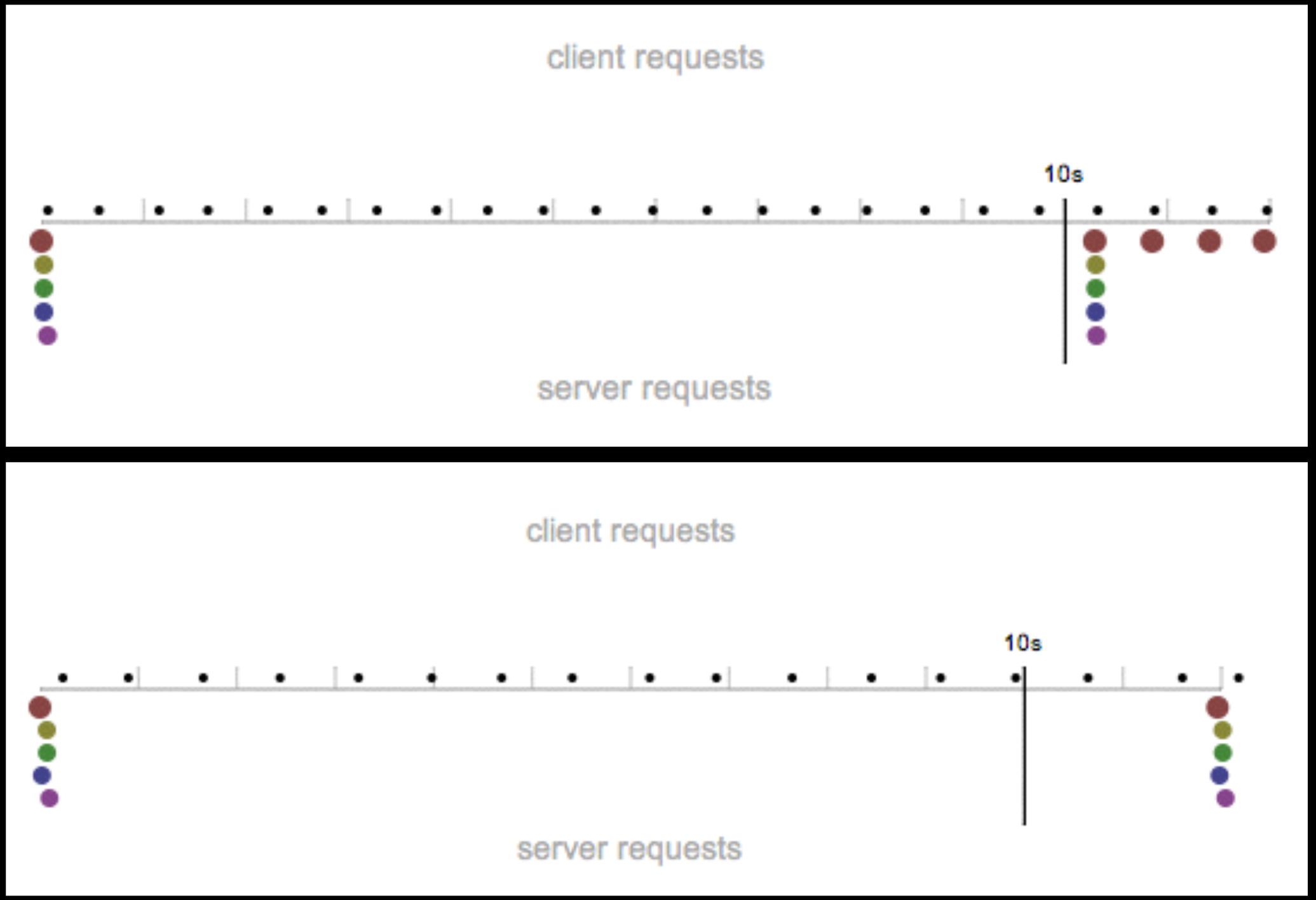
These tests may have unpredictable results if you instructed your browser to force-reload the cache (e.g., by hitting shift-reload or pressing return in the address field), or if there is a proxy cache between your browser and the server. For best results, clear your cache before loading this page.

See this blog entry for more information.

> Testing Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_4; en-au) AppleWebKit/533.18.1 (KHTML, like

max-age / expires

Chrome 6

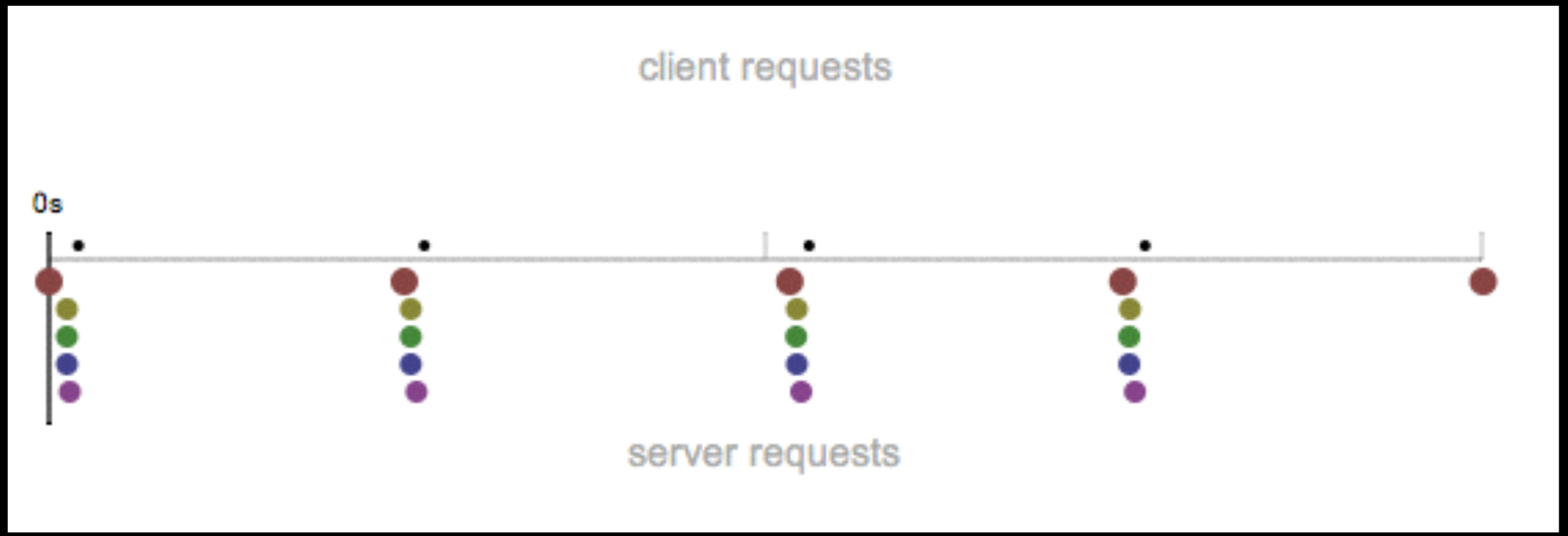


IE7

max-age / expires

Take-away: Cache timing is OK, but not exact

no-cache no-store



Take-away: no-store is enough

Age

Date

Safari 5.0.2

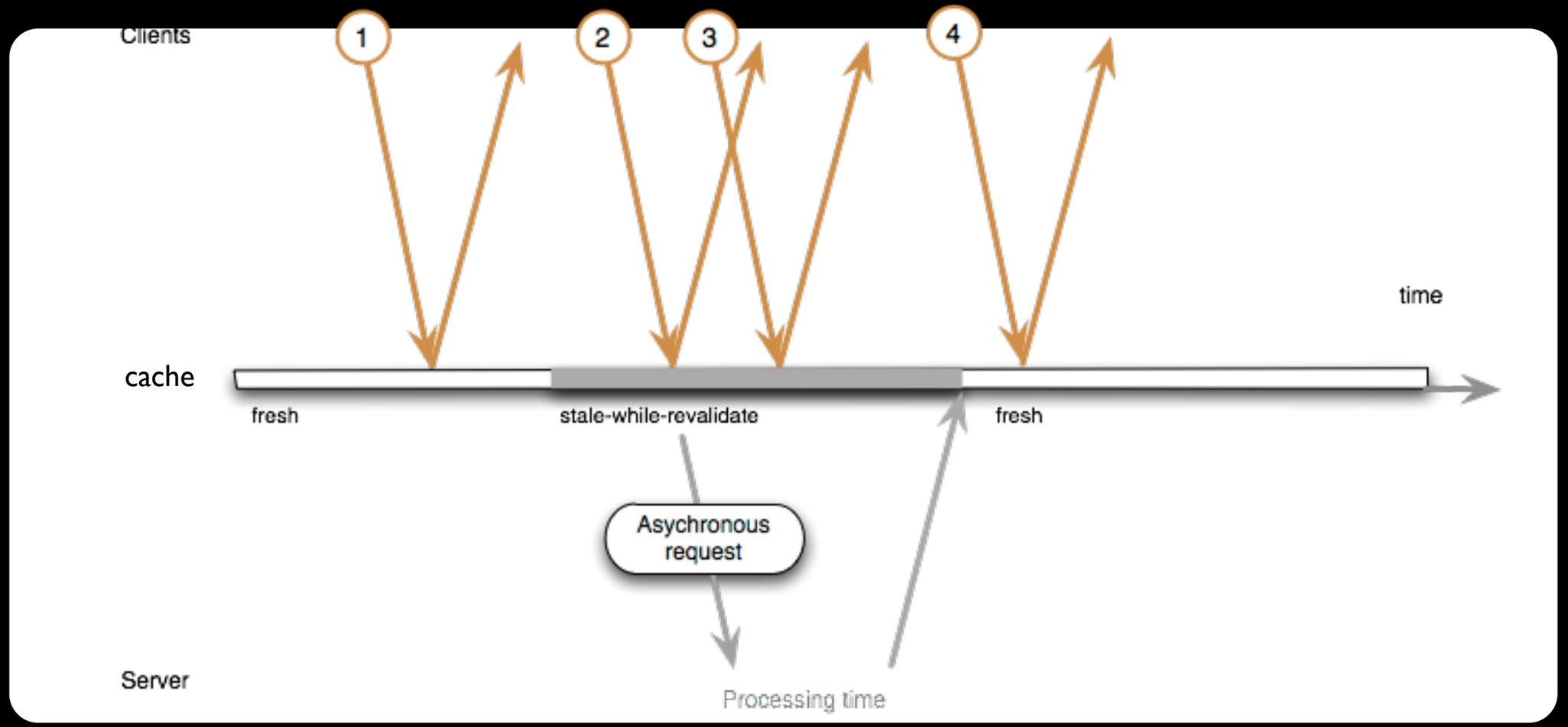
IE (any)

pre-check, post-check

Take-away: max-age=0, pre-check=0, post-check=0 is a really bad idea.

Where is this thing going? (“the talk”)

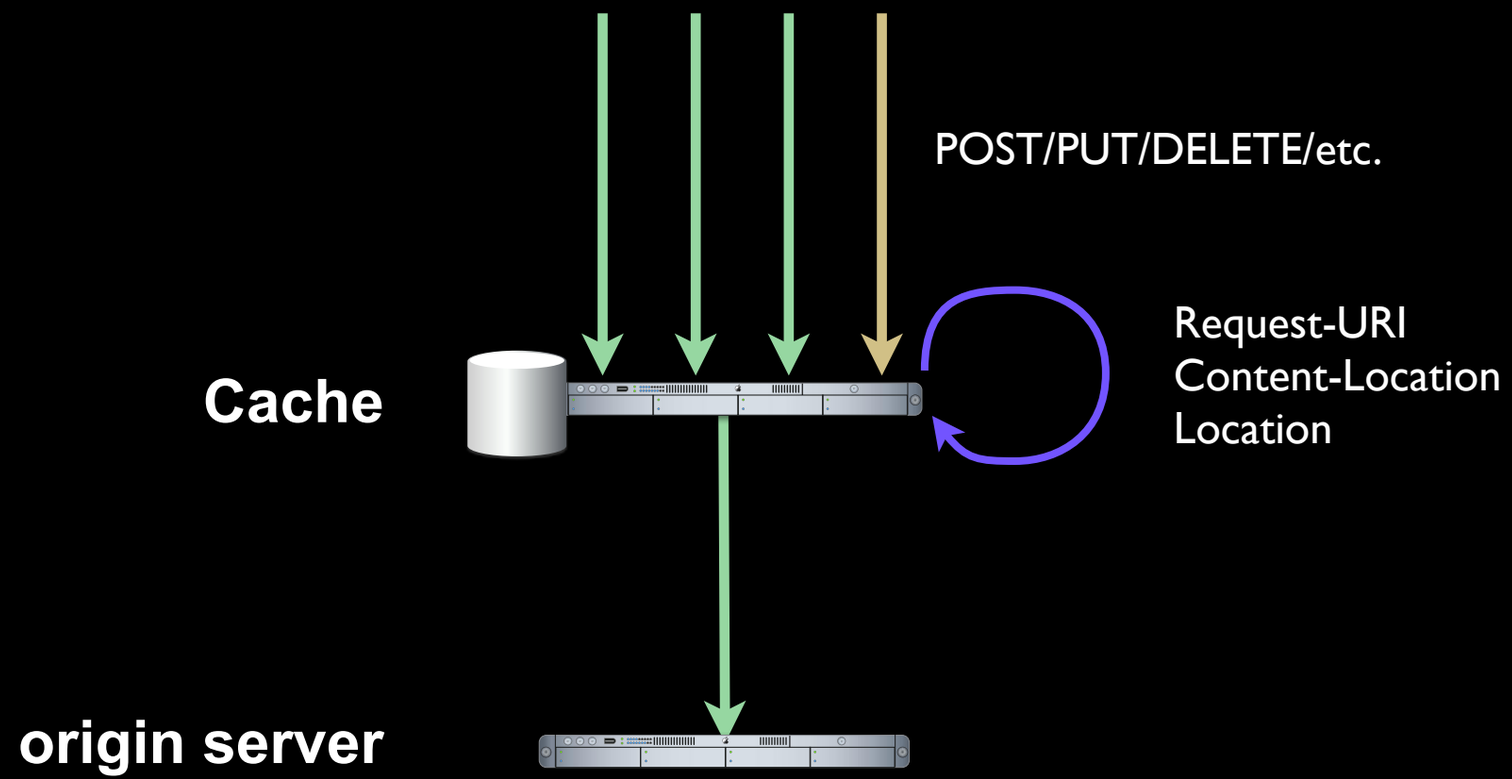
- Fixing Bugs
- Sizing Caches Appropriately
- Replacement Algorithms



stale-while-revalidate

stale-if-error

Cache Invalidation



RFC 2616:

Invalidations after Updates or Deletions

POST /articles/123/new_comment

/newest_comments

/articles/123/comments

/comment_feed

Problem: Related Responses

POST /articles/123/new_comment

/newest_comments

```
Link: </articles/123/new_comment>; rel="invalidat
```

/articles/123/comments

```
Link: </articles/123/new_comment>; rel="invalidated-by"
```

/comment_feed

```
Link: </articles/123/new_comment>; rel="invalidated-by
```

Link: rel=invalidated-by

POST /articles/123/new_comment

/cat/vuvuzela

/newest_comments

```
Link: </articles/123/new_comment>; rel="invalidat
```

/bob/comments

/articles/123/comments

```
Link: </articles/123/new_comment>; rel="invalidated-by"
```

/comment_feed

```
Link: </articles/123/new_comment>; rel="invalidated-by
```

Problem 3: Dynamic Relations

POST /articles/123/new_comment

```
Link: </cat/vuvuzela>; rel="invalidates"
Link: </bob/comments>; rel="invalidates"
```

/cat/vuvuzela

/newest_comments

```
Link: </articles/123/new_comment>; rel="invalidat
```

/bob/comments

/articles/123/comments

```
Link: </articles/123/new_comment>; rel="invalidated-by"
```

/comment_feed

```
Link: </articles/123/new_comment>; rel="invalidated-by
```

Link: rel=invalidates

“side effect” invalidation + link relations =

Linked Cache Invalidation

Further Out
(“there’s still life in this thing”)

- Cache API for invalidation
- Revisiting offline caching
- Forward Cache Digests
- Explicit Cache Key