CS 240
Computer Systems

#21: Caching in HTTP (Age and ETag)

April 7, 2022 · Wade Fagen-Ulmschneider

Q: What is caching?

...have we talked about caching before?

Q: Why is caching important on a cloud-scale?

Caching can occur at multiple different places – and we have different strategies and techniques for each "layer" of the full caching strategy.

[1]:

[2]:

[3]:

Caching Technique #1: Age-Based Caching

Idea + Purpose of Age-Based Caching:

Age-Based Caching Use Case:

- The CS 240 course websites the Bootstrap CSS library from stylizing the front-end content.
- When we load the page from a browser that has never visited the CS 240 website, we see the following requests:

GET https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js

```
1 GET /bootstrap/4.2.1/js/bootstrap.min.js HTTP/1.1\r\n
2 [...]
```

HTTP Response:

```
1 HTTP/1.1 200 OK\r\n
2 date: Wed, 06 Apr 2022 18:37:00 GMT\r\n
3 age: 1883002\r\n
4 cache-control: public, max-age=31919000\r\n
5 cdn-cache: HIT\r\n
6 cdn-cachedat: 2021-06-08 14:31:34\r\n
… [...]
```

[Line 3]: The `age` header:

[Line 4]: The `cache-control["max-age"]` header:

Q: What happens when we request a page using this resource again before the content expires?

....how much traffic is saved?

...how much time is saved?

...what are disadvantages of age-based caching?

Caching Technique #2: Entity Tag Caching
Idea + Purpose of Entity Tag (ETag) Caching:

ETag-Based Caching Use Case:

- Some content is likely to be the same for a period of time, but change occasionally and at unpredictable times.
- Ex: My Institute for Interactive Visualization Systems site has data updates to the visualizations each day. Here's a request for the snow data in Champaign, IL:

GET https://vis.cs.illinois.edu/data/weather-snow/USC00118740.csv

```
1 GET /data/weather-snow/USC00118740.csv HTTP/1.1\r\n
2 [...]
```

HTTP Response:

```
1 HTTP/1.1 200 OK\r\n
2 age: 12278
3 date: Wed, 06 Apr 2022 15:30:13 GMT\r\n
4 etag: "aa7cc2cf9f22e2dc3bc9e5e1afcf150f"\r\n
5 x-cache: Hit from cloudfront\r\n
… [...]
```

[Line 4]: The `etag` header:

Q: What happens when we request a page again?

GET https://vis.cs.illinois.edu/data/weather-snow/USC00118740.csv

```
1 GET /data/weather-snow/USC00118740.csv HTTP/1.1\r\n
2 If-None-Match: "aa7cc2cf9f22e2dc3bc9e5e1afcf150f"\r\n
  [...]
```

HTTP Response:

```
1 HTTP/1.1 304 Not Modified\r\n
… [...]
```

….how much traffic is saved?

…how much time is saved?

Q: What are the disadvantages of Entity Tag caching?

One Additional, Modern Drawback:
In recent years, regulations have prohibited various tracking using a specific technology called HTTP Cookies. If a developer can't use a cookie, what else can they use to track you?

Example:

GET /trackingImage.png

```
1 GET /trackingImage.png HTTP/1.1\r\n
2 [...]
```

HTTP Response:

```
1 HTTP/1.1 200 OK\r\n
2 etag: "unique-identifier-just-for-you"\r\n
… [...]
```

When you visit again:

GET /trackingImage.png

```
1 GET /trackingImage.png HTTP/1.1\r\n
2 If-None-Match: "unique-identifier-just-for-you"\r\n
… [...]
```

Your Web Application
Your web application generates a complex response for every request. Your AWS server can handle **10 requests /second** when you must generate a new response and the cost of it is exactly $4 /month. *(You can assume you can return 1,000+ cached responses /second.)*

Q1: You need to set up a cloud-scale system that handles 1,000 requests a second. How many EC2 instances do you need if you use no caching? What is the cost?

Q2: If you have a cache-hit rate of 70%, how does that change things?

…how does ETag and Age-based differ here?