CS 240
Computer Systems

#20: HTTP Caching

Nov. 4, 2021 · Wade Fagen-Ulmschneider

Directory Permission Bits

Following up from lecture on Tuesday, what impact does directory read and execute bits have? Here's my experiment:

- Create two directories: `test-no-r` and `test-no-x`.
- Create one file in each directory, `hello.txt`.

The directories with permission bits set:

```
$ ls -la
d-wx--x--x 2 waf  waf  4096 Nov  3 11:20 test-no-r
drw-r--r-- 2 waf  waf  4096 Nov  3 11:20 test-no-x
```

Testing the `ls` command to list files:

```
$ ls test-no-r
ls: cannot open directory 'test-no-r': Permission denied
```

```
$ ls test-no-x
ls: cannot access 'test-no-x/hello.txt': Permission
denied
hello.txt
```

Testing the `cat` command to list contents of the known file:

```
$ cat test-no-r/hello.txt
Hello world!
```

```
$ cat test-no-x/hello.txt
cat: test-no-x/hello.txt: Permission denied
```

Conclusion:

- `r` permission bit on directories:

- `x` permission bit on directories:

Caching

Caching is critical across all parts of computer systems. We have already seen two forms of caching already:

1.

2.

Caching with ETags in HTTP

The HTTP protocol has caching built in at the protocol layer! Nearly every HTTP request you make will have a response returned with cache-specific tags:

HTTP Request without cached data:

```
GET /lecture.jpg HTTP/1.1\r\n
[...]
```

HTTP Response

```
HTTP/1.1 200 OK\r\n
Date: Wed, 03 Nov 2021 16:31:20 GMT\r\n
Last-Modified: Tue, 01 Sep 2020 17:07:47 GMT\r\n
ETag: "8073356a8280d61:0"\r\n
Content-Length: 25725\r\n
[...]
```

Future requests will refer to the ETag to determine if the contents of the file has changed:

HTTP Request with cached data stored locally:

```
GET /lecture.jpg HTTP/1.1\r\n
If-None-Match: "8073356a8280d61:0"\r\n
[...]
```

HTTP Response on cache hit:

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

Q: If you visit a webpage 100 times a day, how many times would you need to check the ETag?

HTTP Cache Alternative: Age-Based Cache Policy

Instead of using a cache tag, another HTTP strategy is to accept the maximum allowed age of a file:

HTTP Response

```
HTTP/1.1 200 OK\r\n
Cache-Control: public, max-age=31919000\r\n
Age: 6745054\r\n
[...]
```

The age and max-age are specified in ___________________:

- **max-age**: 31919000 __________ = ____________________
- **age**: 6745054 __________ = ____________________

Q: If you visit a webpage 100 times a day, how many times would you need to request the cached file?

ETag Caching	Age-Based Caching
Best Used For:	Best Used For:
Drawbacks:	Drawbacks:

Caching Efficiency

In cloud systems, one of the most motivating factors is cost. Consider the bandwidth cost for AWS EC2 instances:

Usage Category	Cost
0 GiB - 1 GiB	$0.00 *First GiB is free!*
1 GiB - 10 TiB	$0.09 per GiB

https://aws.amazon.com/ec2/pricing/on-demand/

Suppose you're running a website that sends 100,000 HTTP packets /day where your average packet headers of 0.1 KiB and the content is 200 KiB.

1. How much bandwidth would be used in a 31-day month?

2. How much would that bandwidth cost on AWS?

3. You implement ETag caching and you find that your server has a cache-hit rate of 50%. How much bandwidth and money would you save?

4. You implement age-based caching and you find that your server has a cache-hit rate of 50%. How much bandwidth and money would you save?