

Cache strategies

Vitalii Melnychuk

"There are only two hard
problems in Computer Science:
cache invalidation and naming things."

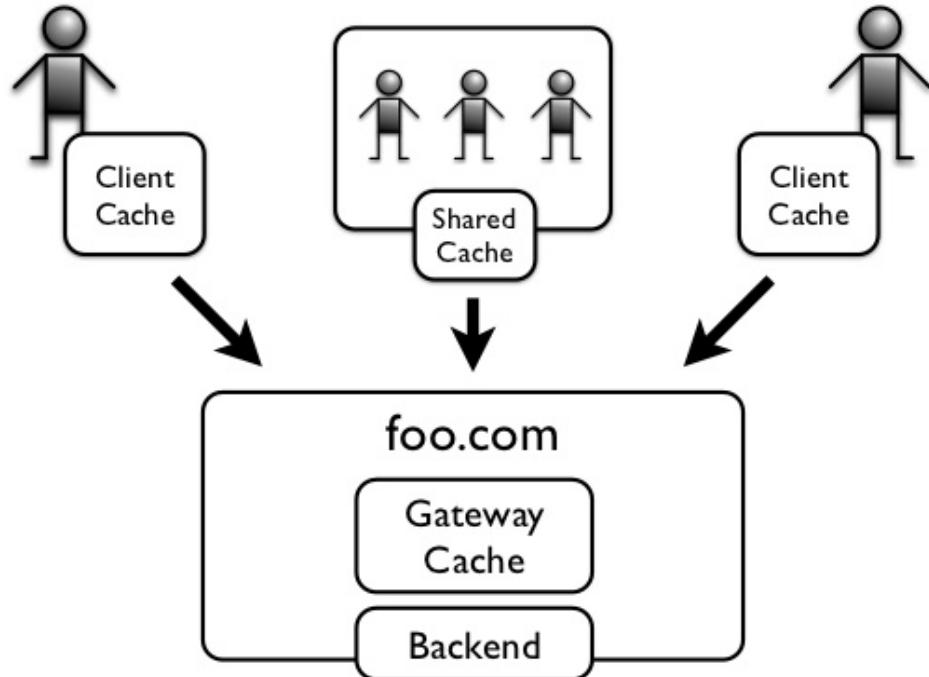
Phil Karlton

Is it important to use cache?

A scene from Toy Story featuring Woody and Buzz Lightyear. Woody, on the left, has a serious expression and is looking upwards. Buzz, on the right, is smiling and pointing his right arm towards the sky. They are standing in a room with a chalkboard in the background.

CACHE

CACHE EVERYWHERE



- decorative
- information
- text can be instead



Within STB Website

Go

Singapore
TOURISM
BOARD

About STB

Newsroom & Publications

Careers

Invest in Tourism

 I am a... ▾

Other STB Websites

Tourist Information

STATISTICS &
MARKET INSIGHTS

INDUSTRIES

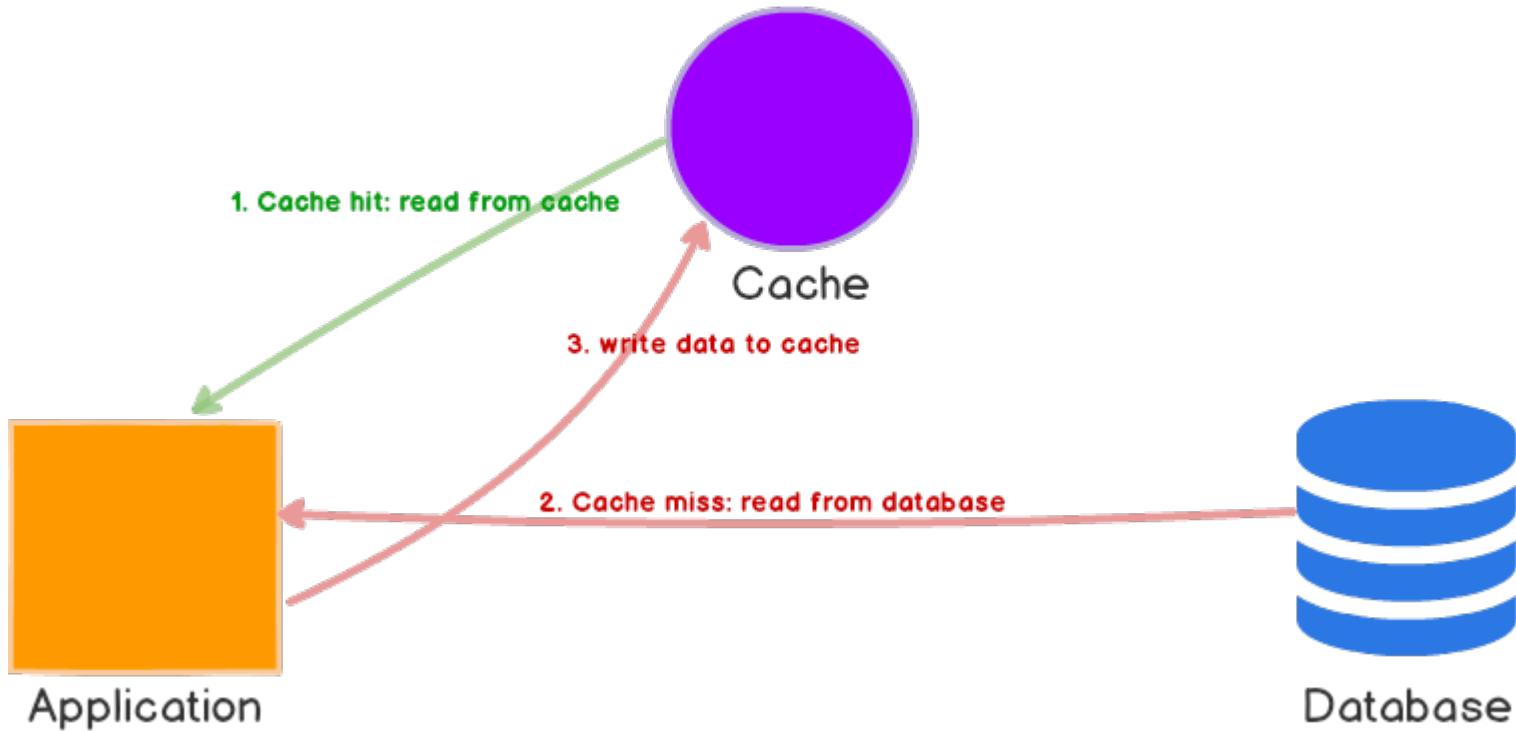
ASSISTANCE
& LICENSING

TRADE EVENTS &
RESOURCES

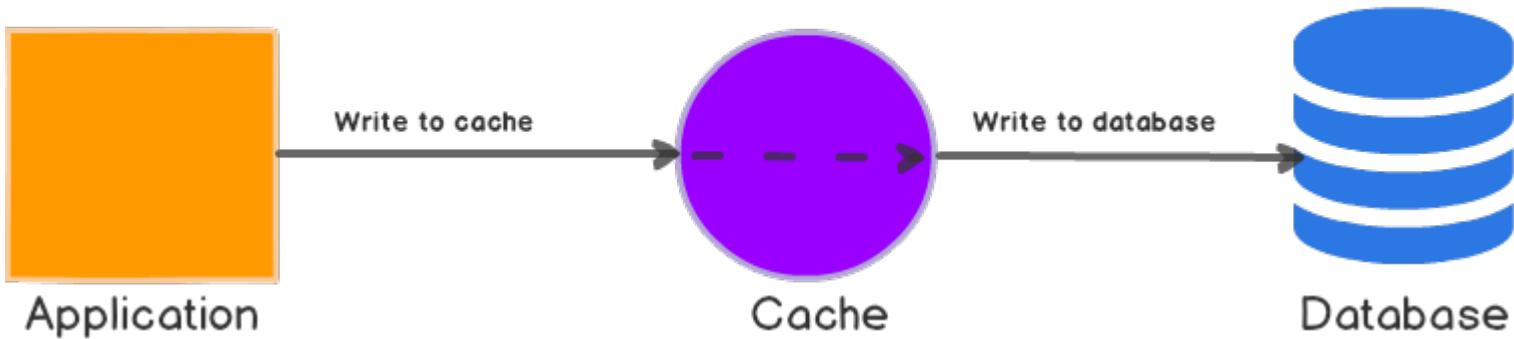


New Brand Launch

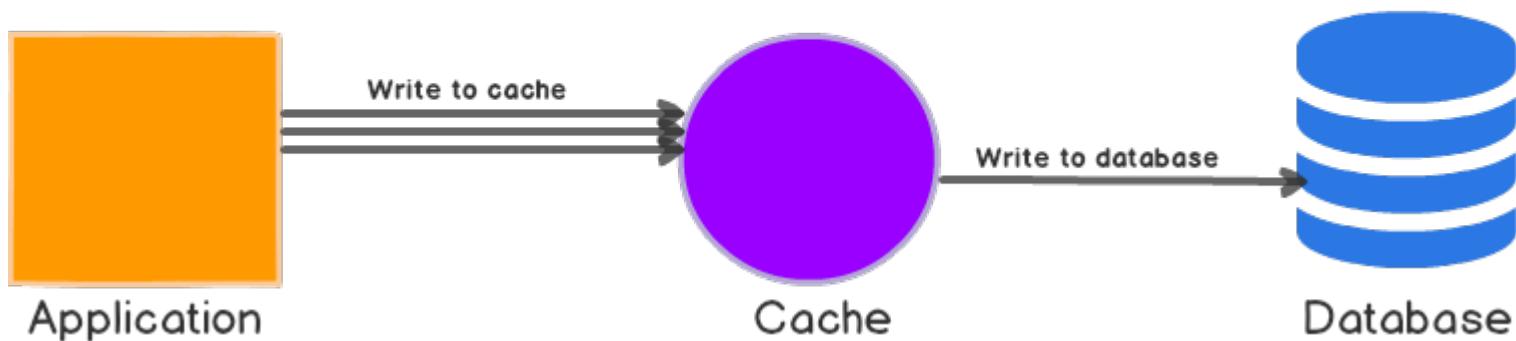
Cache-Aside



Write-Through Cache



Write-Back



Cache invalidation

Cache Lifetime

Advantages:

- Easy to implement
- Easy to manage
- Without additional data

- Data is not persistent

Disadvantages:

HELLO, FRONTEND

**HAVE YOU TRIED CLEARING
YOUR CACHE**

```
#crontabfi
```

```
*/5 * * * * ./bin/clear-cache.sh
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~
```



HELLO IT

**HAVE YOU TRIED TURNING IT OFF
AND ON AGAIN?**

By using Tags

Advantages:

- Data is persistent

- Tags management

Disadvantages:

Naming things

user-1: {"auth":true}

account-35: {"settings-resources":["mails"]}

monitoring: {"dates":[1-05-2017]}

beta: true

users-count: 5000

dashboard-data:
{"some-key":12345}

uke::users-count

duke::guest

duke::guest::register

duke::guest::login

duke::guest::register::show-register-button

duke::account-1

duke::account-1::user-500

duke::account-1::user-500-log-data

duke::account-1::user-234

```
$ redis-cli --raw keys "$PATTERN" | xargs redis-cli del
```



You can't do this with one request only.

1

But you can loop on every keys using the same prefix to delete all of them like so :



```
$cache = new Memcached();
$keys = $cache->getAllKeys();
$prefix = "always_this_one_";
$prefixLength = strlen($prefix);

foreach ($keys as $key) {
    if (substr_compare($key, $prefix, 0, $prefixLength) === 0) {
        $cache->delete(substr($key, 0, -1));
    }
}
```

Questions?