

6 Ways to Get Cached

Speeding up your web app and simplifying life



Paweł Komarnicki!

@pawelkomarnicki

www.komarnicki.webd.pl

Cache? Why?

Typical website

mostly reads + some writes

Typical application

mostly reads + some writes +
lots of calculation

read calculate write

read calculate write

read

calculate

write

CPU gets „too hot”

complexity = high cost

Database = slow

big results = lots of RAM

retrieving = time cost

Page load time

generation + transfer



Cache FTW!

Backend

= generation time

View caching

- page caching
- action caching
- fragment caching

View caching

- Overall idea
 - generate the page or fragment
 - then store it as an HTML static file
 - sometime in the future invalidate

View caching

- page caching
= one cache for the whole page
- action caching
= one cache based on filters
- fragment caching
= multiple caches for one page

Class caching

```
def current_user  
  User.find session[:user_id]  
end
```



```
def current_user  
  @current_user ||=  
    User.find session[:user_id]  
end
```

Data caching

class < data < view



memcached

fast

flexible

not persisted

```
# config/environments/development.rb
config.cache_store = :mem_cache_store

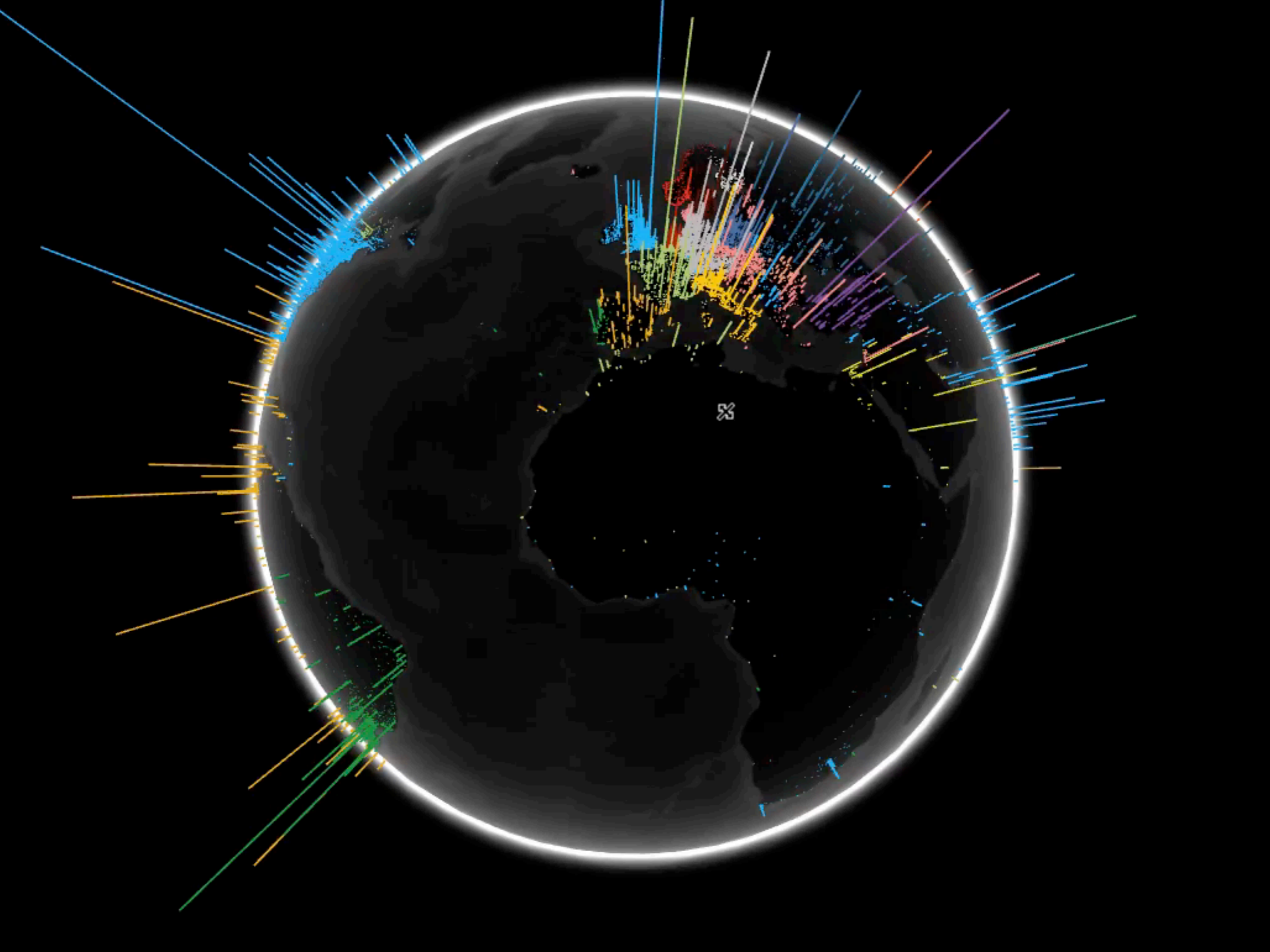
# model or controller method body
Rails.cache.fetch("cache_key") do

  # Perform some really heavy calculations
  User.all.collect { |user|
    user.calculate_time_spent_in_website
  }

end

# write = store (overwrite)
# fetch = read (and write if empty)
# read = read the value
```

**Why data caching
rocks?**



Number of records: 10605

Calculation time: ~ 45 seconds

max $360 * 180 = 64800$ points

maintenance > invalidation

Frontend
= transfer time

no access = no problem

HTTP Headers

- `max-age`
- `etags`
- `last_modified`

Cache-Control: max-age=28800

= „I won't change for the next 8 hours”

instantly loading

no invalidation

ETag: "canihazcheezburgerz"

= „My current version has this etag, store it!”



If-None-Match: "canihazcheezburgerz"

= „My last version has this etag, am I still good?”

can be invalidated

hits the server

Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT

= „I was modified last time on 15 Nov ..”

Works like ETag, but means byte-to-byte identical file (*etag = semantically identical*)

can be invalidated

hits the server

Asset caching

- packaging
- css sprites
- cache manifest

Asset packaging

- merge scripts into packages
- minimise, gzip, uglify etc.
- 1 file = faster download

CSS sprites

- merge lots of small images into one
- faster download

Cache manifest

- offline access to assets
= main use case
- mentioned files are
cached in the browser

Remember to include it in the page:

```
# <html manifest="clock.appcache">
```

CACHE MANIFEST

All files will be loaded from the web server and stored in the browser cache:

/main/home

/main/app.js

/settings/home

/settings/app.js

Cool feature: we can also cache external sources (JavaScript libraries?)

<http://img.example.com/logo.png>

<http://path.to.jquery/script.js>

Local Storage

- native localStorage
- JSON storage libs like Lawnchair

```
// Save data like into any key-value store  
localStorage.setItem(  
    "username", "Paweł"  
);
```

```
// Access some stored data  
console.log(  
    localStorage.getItem("username")  
);
```

```
var store = new Lawnchair(  
  {name: 'testing'},  
  function(store) {  
  
    // Create an object  
    var me = {key: 'Paweł'};  
  
    // Save it  
    store.save(me);  
  
    // Access  
    store.get('Paweł', function(me) {  
      console.log(me);  
    });  
  });  
});
```

Demo

SPORT SESSIONS



Add Workout

September 2011		October 2011					November 2011	
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday		
26	27	28	29	30	1	2		
3	4	5	6	7	8	9		
10	11	12	13	14	15	16		
17	18	19	20	21	22	23		
24	25	26	27	28	29	30		
30	31	1	2	3	4	5		

Use advanced filtering

GOLD

DATE

2011 179

October 5

September 7

August 32

July 12

June 21

May 35

April 28

March 20

February 9

January 10

2010 234

2009 32

SPORT TYPE

Running 93

Date	Type	Distance	Duration	Speed Avg.	Pace	kCal	HR Av
10/03/2011	Race Cycling	6.47 km	15:28	25.08 km/h	2:23 min/km	122 kcal	
10/03/2011	Race Cycling	6.51 km	15:53	24.59 km/h	2:26 min/km	121 kcal	
10/06/2011	Cycling	6.24 km	19:20	19.37 km/h	3:05 min/km	133 kcal	
10/06/2011	Cycling	6.46 km	19:26	19.93 km/h	3:00 min/km	140 kcal	
10/09/2011	Strolling	4.03 km	56:34	4.27 km/h	14:02 min/km	253 kcal	
Overall statistics		29.71 km	2:06:43			769 kcal	

SPORT SESSIONS



Add Workout

< September 2011 October 2011 November 2011 >						
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
30	31	1	2	3	4	5

Use advanced filtering

GOLD

DATE

2011 179

October 5

September 7

August 32

July 12

June 21

May 35

April 28

March 20

February 9

January 10

2010 234

2009 32

SPORT TYPE

Running 93

Date	Type	Distance	Duration	Speed Avg.	Pace	kCal	HR Av
10/03/2011	Race Cycling	6.47 km	15:28	25.08 km/h	2:23 min/km	122 kcal	
10/03/2011	Race Cycling	6.51 km	15:53	24.59 km/h	2:26 min/km	121 kcal	
10/06/2011	Cycling	6.24 km	19:20	19.37 km/h	3:05 min/km	133 kcal	
10/06/2011	Cycling	6.46 km	19:26	19.93 km/h	3:00 min/km	140 kcal	
10/09/2011	Strolling	4.03 km	56:34	4.27 km/h	14:02 min/km	253 kcal	
Overall statistics		29.71 km	2:06:43			769 kcal	



Thank you!

Questions?