



# *Анализ и оптимизация на софтуерни приложения*

Александър Пенев

Васил Василев

## Оптимизация на WEB приложения

# Съдържание

- ❖ Обща схема на работа на едно WEB приложение
- ❖ Кеш на различни нива (browser, proxy, server)
- ❖ HTTP
- ❖ Инструменти за benchmark и други

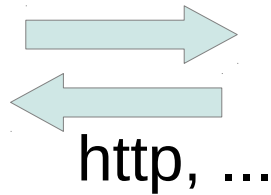


# Класическа схема на трислойно WEB приложение

Browser /  
Web Клиент



*Firefox, Chrome,  
Safari, IE, ...*

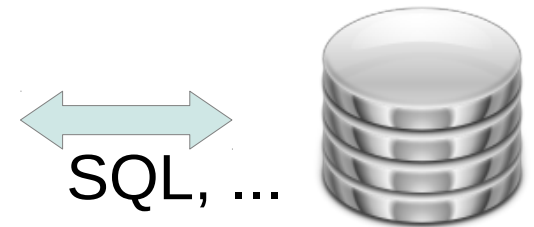


Web Сървър



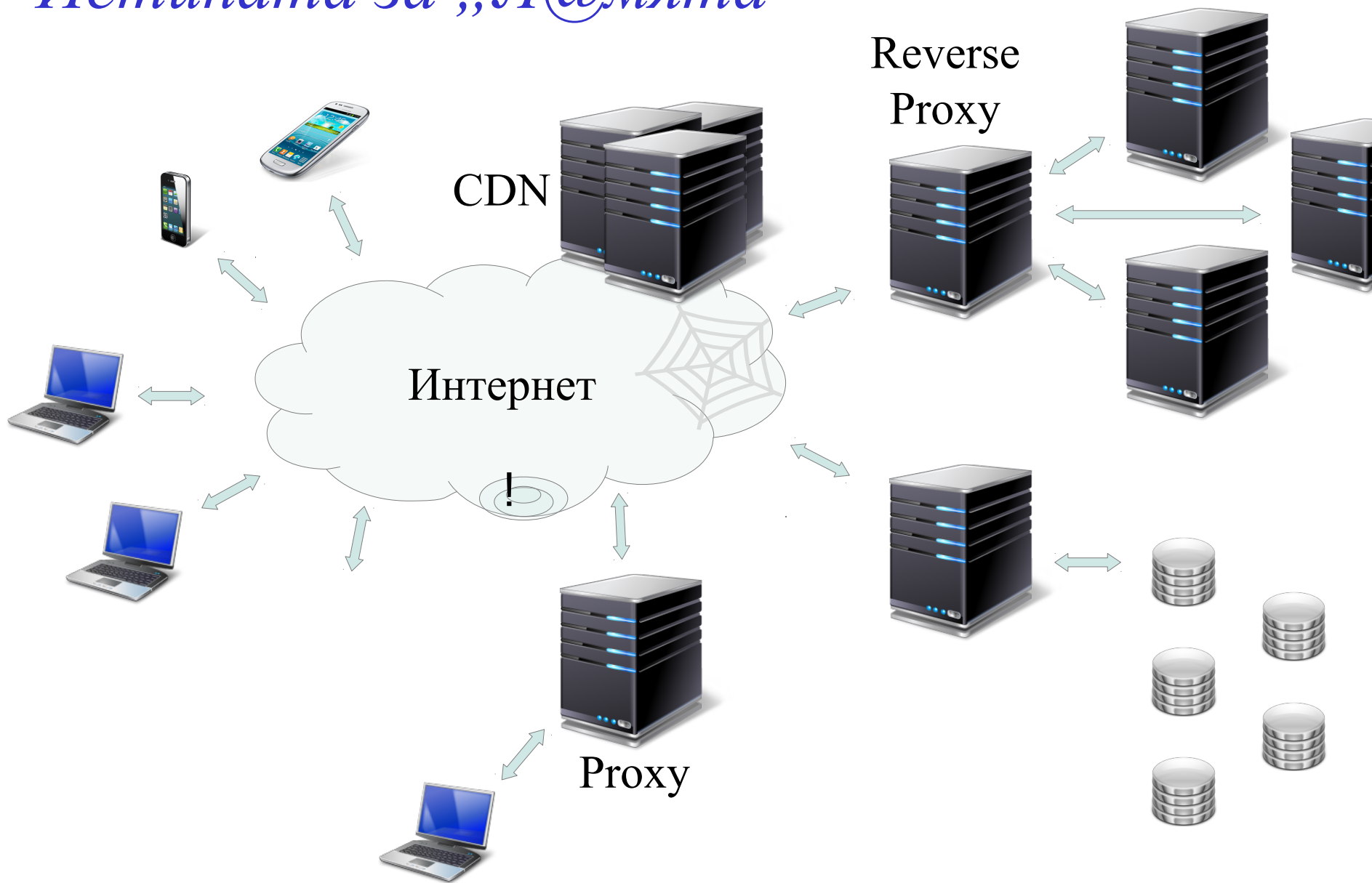
*Apache, IIS, ...*

База данни /  
Файлове ...



*MySQL, MSSQL,  
PostgreSQL,  
Oracle, Firebird, ...*

# Истината за „Л@мята“



# Как може да ускоряваме?

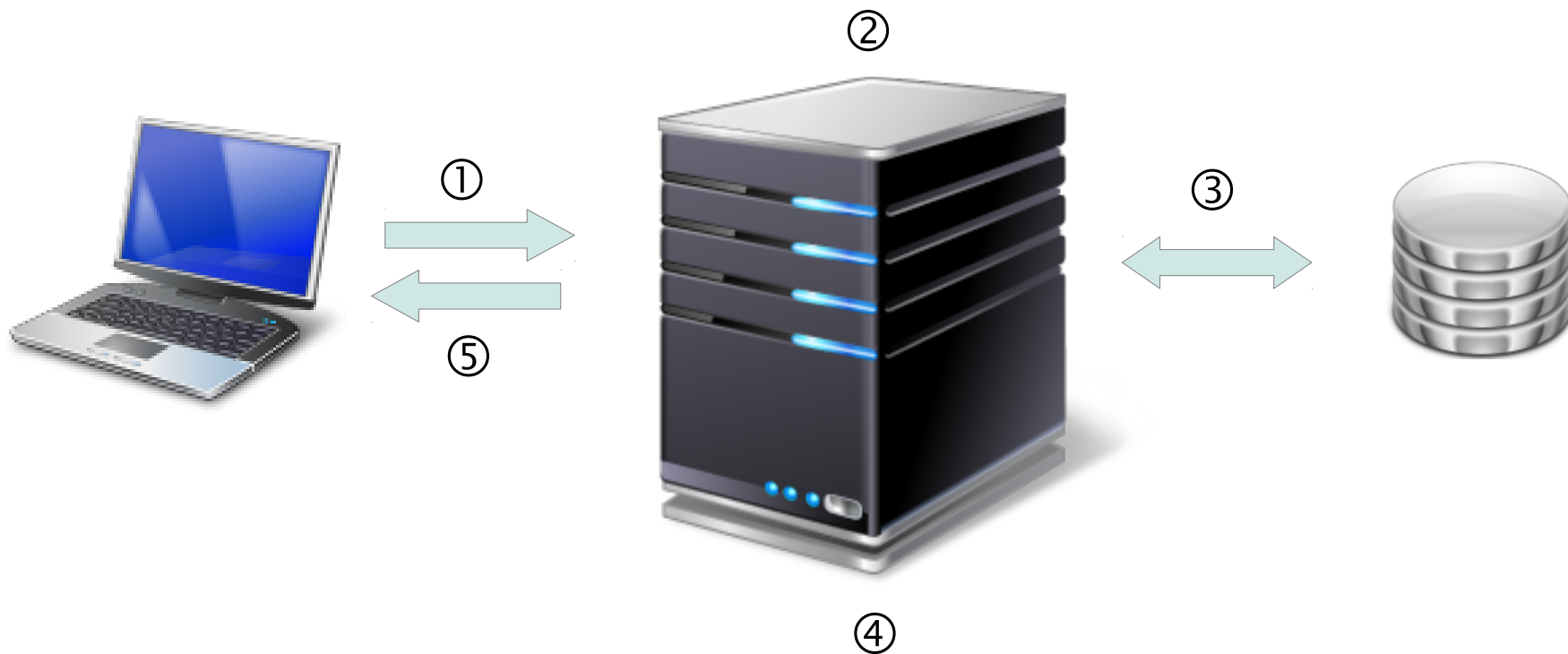
- ❖ **Файлов/Ресурсен Кеш във Web клиента (браузъра)**  
*Най-бърз достъп – няма мрежова комуникация. Кеш контрол.*
- ❖ **Прoxy Кеш (междинни кеширащи сървъри)**  
*Кеш контрол (expires, ...). Съхранява на ниво организация или ISP, CDN, https, ...*
- ❖ **Кеш във Web сървъра (Application кеш)**  
*При възможност намалява използваните изчислителни ресурси на ниво сървър (за сметка на диск и памет). Изпълнение на скриптове, заявки до база данни и др.*
- ❖ **Минимизиране на броя на HTTP заявките**  
*Всяка Интернет страница се състои от множество ресурси (html, jpg, css, js). Всеки ресурс се извлича с отделна http заявка, а те са „скъпи“. KeepAlive. CSS Sprites. AJAX. HTTP 2.0?...*
- ❖ **Компресия / Намаляване на размера на данните**  
*Компресията и други техники (JS/CSS Minify, HTTP-304) помагат да се пести „най-скъпия ресурс“ на Web приложенията – мрежовата комуникация (пропускателна способност и латентност на канала).*
- ❖ **Увеличаване на ресурсите**  
*Мащабируемост при много едновременни клиенти, големи обеми от данни и др.*



# Кеш на различни нива



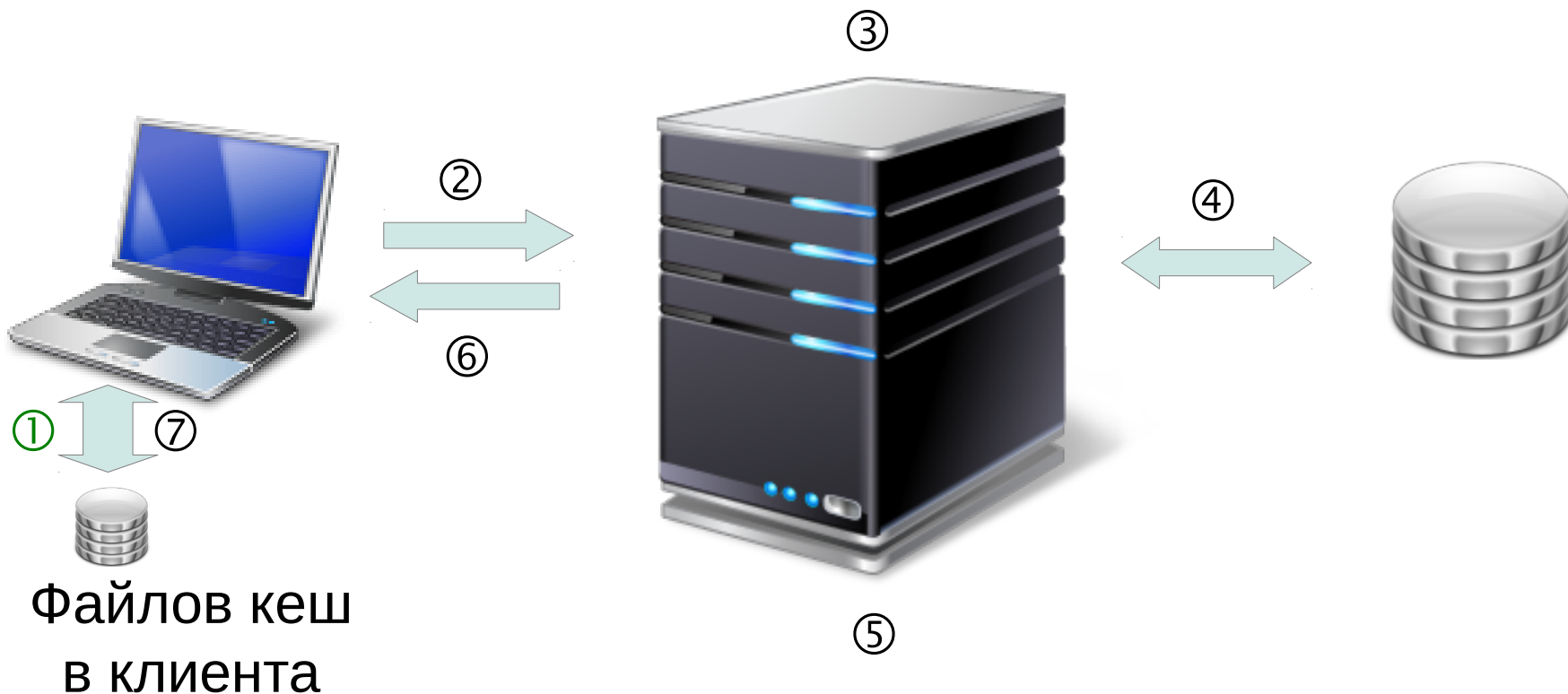
# Начална ситуация (без Кеши)



# Подобряване на производителността Файлов/Ресурсен Кеш във Web клиента

Hit: ①

Miss: ① ② ③ ④ ⑤ ⑥ ⑦

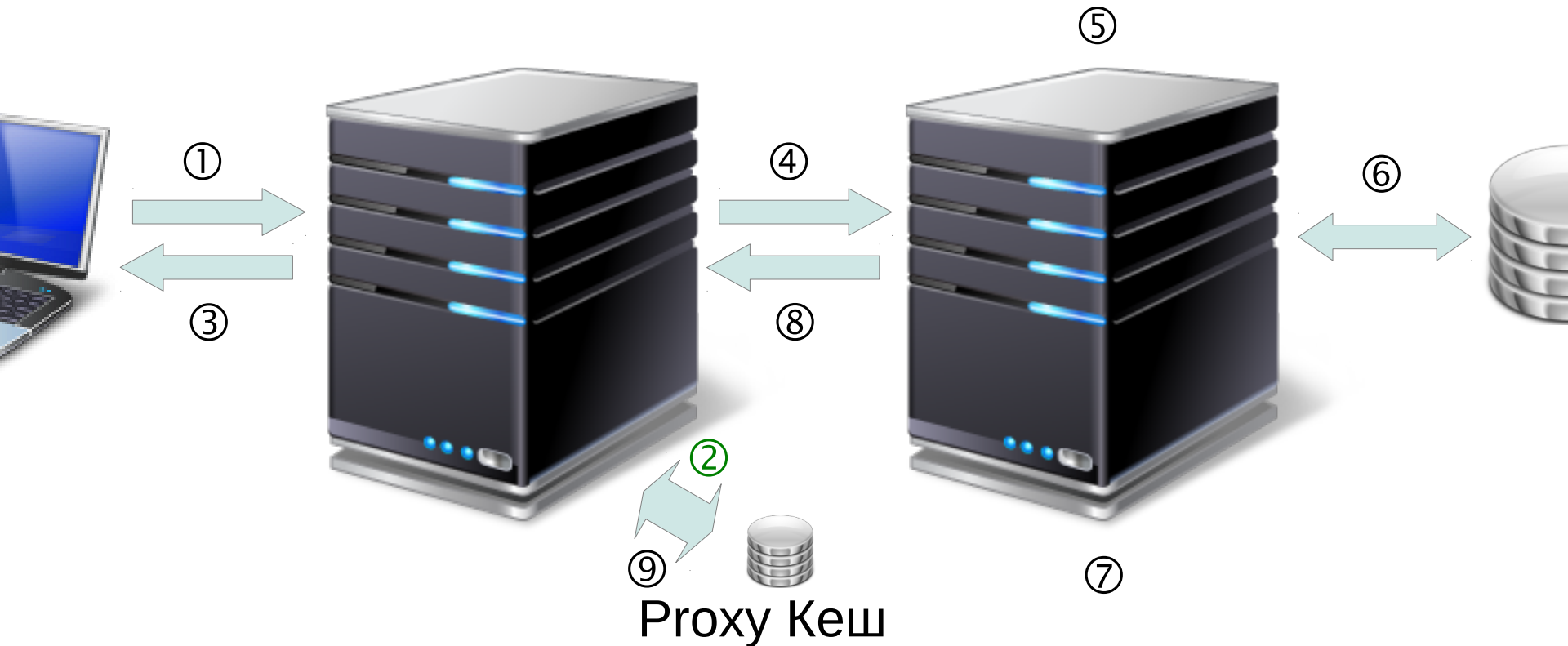




# Подобряване на производителността Proxy Кеш

Hit: ① ② ③

Miss: ① ② ④ ⑤ ⑥ ⑦ ⑧ ⑨ ③

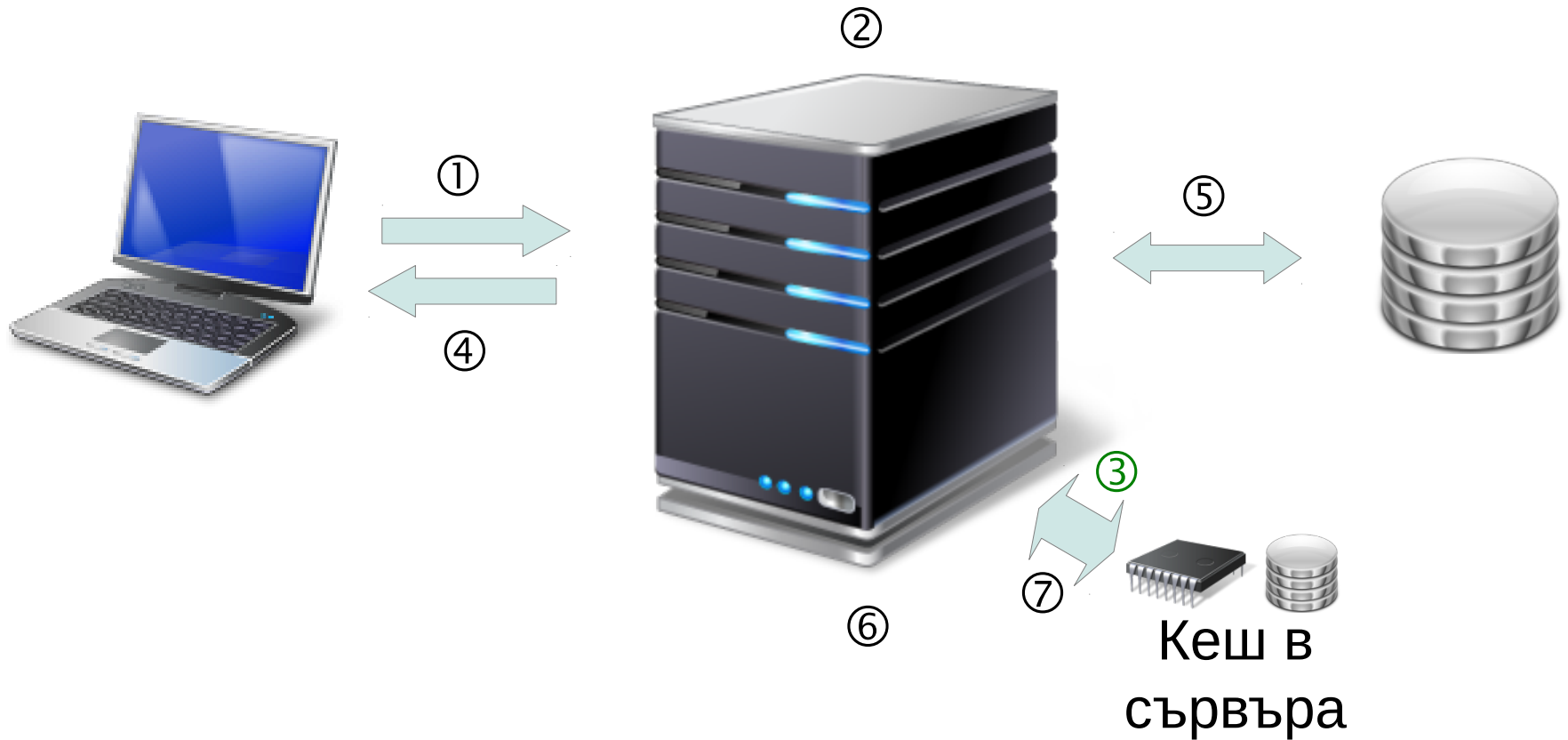


# Подобряване на производителността

## Кеш във Web сървъра

Hit: ① ② ③ ④

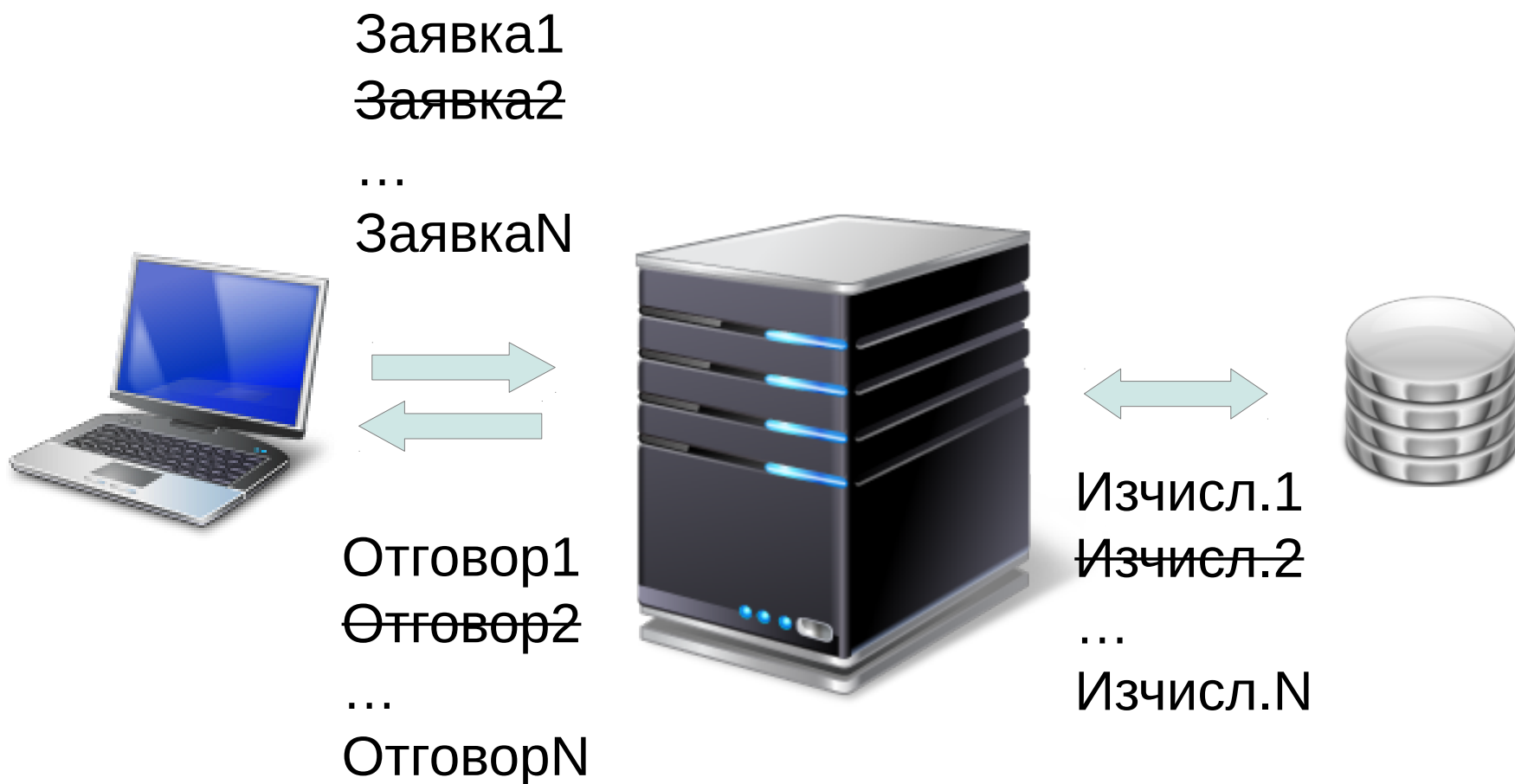
Miss: ① ② ③ ⑤ ⑥ ⑦ ④



# Намаляване на броя на HTTP заявките

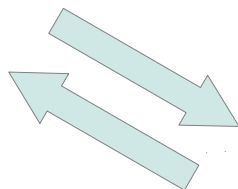
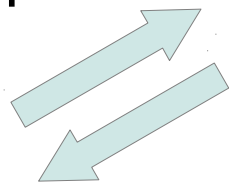


# HTTP заявки към сървъра (сървърите)



# Едновременна заявка към повече от един HTTP сървър

.html,  
.php, .aspx



.png, .jpeg,  
.js, .css

# Компресия / Намаляване на размера на данните



# Компресия (цел да се намали трансфера)



# Компресия (предварителна)





# Кеш контрол

## HTTP Response 200 (OK)

Има ли  
по-нов  
файл?

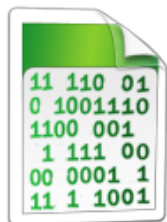
Да



URL

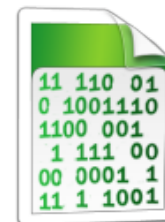
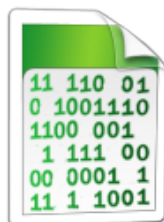


HTTP 200



Кеш

(15.02.2014)



Сървър

(28.02.2014)



# Кеш контрол

## HTTP Response 304 (Not Modified)

Има ли  
по-нов  
файл?

Не



URL

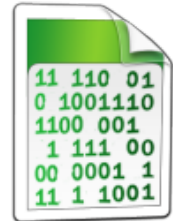


HTTP 304



Кеш

(23.02.2014)



Сървър

(23.02.2014)



# Увеличаване на ресурсите



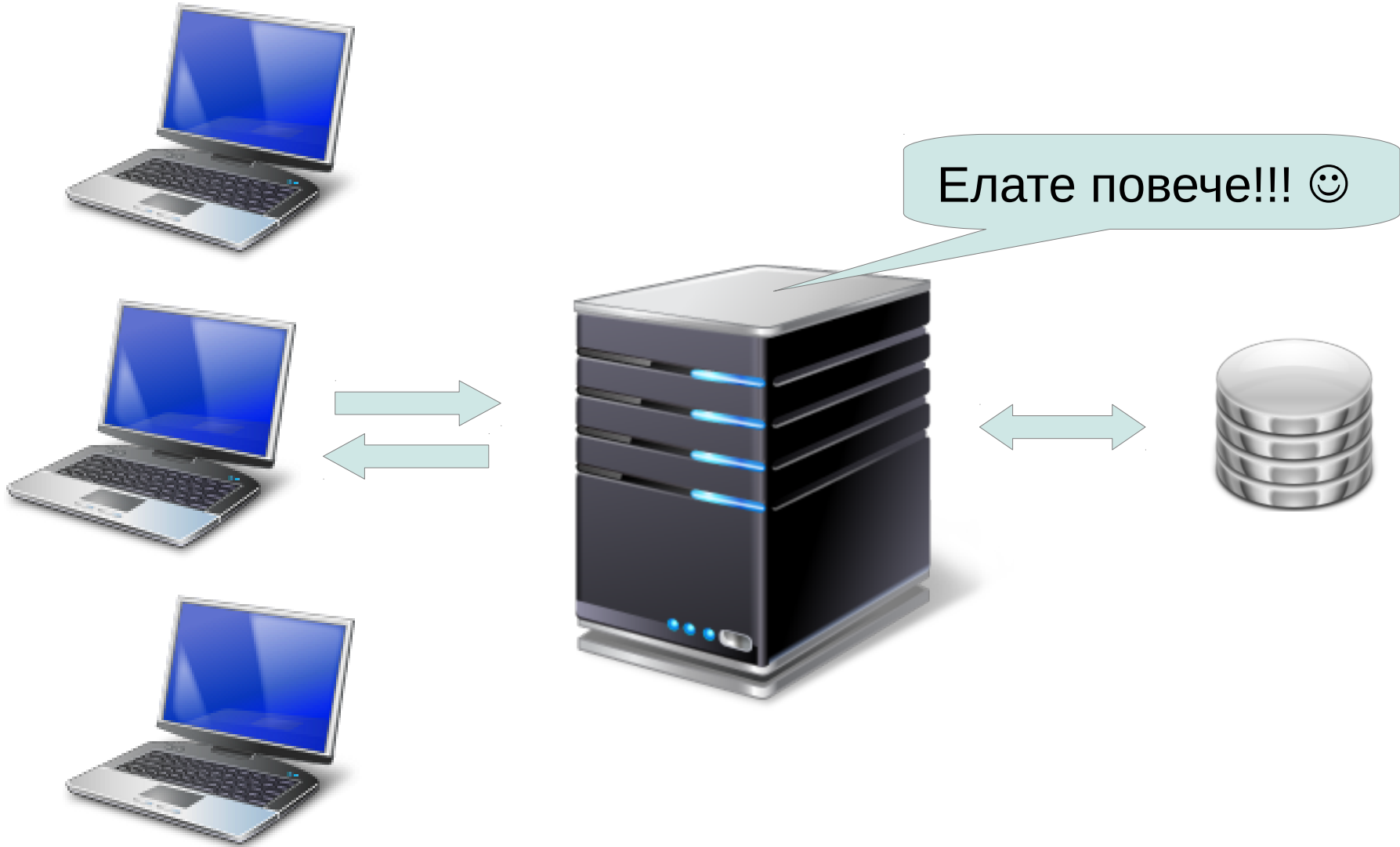
# Подобряване на производителността

## Повече сървъри, бази данни, ресурси ...



# Подобряване на производителността

## Повече сървъри, бази данни, ресурси ...



# Подобряване на производителността

## Повече сървъри, бази данни, ресурси ...



# Подобряване на производителността

## Повече сървъри, бази данни, ресурси ...



# Подобряване на производителността

## Повече сървъри, бази данни, ресурси ...





# Инструменти

- ❖ Firebug, Dragonfly, Web Inspector и други аналогични инструменти за разработчици
- ❖ YSlow, Speed Tracer и аналогични
- ❖ ApacheBench (ab) и аналогични
- ❖ Httperf
- ❖ Google Analytics
- ❖ И много други...

