Kešování webu pro vyšší výkon

Petr Krčmář

6. října 2019

O mně

- linuxák od roku 1998
- správce serverů
- školitel
- šéfredaktor Root.cz
- člen vpsFree.cz
- organizátor LinuxDays
- můj web je petrkrcmar.cz

Prezentace už teď na webu

https://www.petrkrcmar.cz

ROOT.CZ vpsFree.cz

Obsah přednášky

1. Co je to keš?
2. Co a kde kešujeme
3. HTTP hlavičky
4. Strategie kešování
5. Nginx jako kešující reverzní proxy

Co je to keš?

Keš je, když...

- mezipaměť pro ukládání webového obsahu
  - HTML, obrázků, CSS, JS...
- data procházejí a mohou být ukládána
- při příštím požadavku může být obsah vydán z keše
- automatické vyprazdňování keše podle různých strategií

ROOT.CZ vpsFree.cz

Proč chcete kešovat

- opakované použití připraveného obsahu
- snížení latence při načítání obsahu
  - stránka je sestavená a může být blíž uživateli
- snížení zátěže na web serveru
  - nemusíme sestavovat stejnou stránku opakovaně
  - i keš s krátkou dobou platnosti může pomoci
- možnost rozložení zátěže na více backendů
- možnost terminace spojení (HTTPS) na frontendu
- překlenutí krátkodobých výpadků

Terminologie

Origin server	webserver schopný generovat obsah
Reverse proxy	předřazený webserver bez vlastního obsahu
Hit ratio	poměr stránek odbavených z keše (více = lépe)
Freshness	informace o aktuálnosti kešovaného obsahu
Validation	ověření aktuálnosti obsahu proti zdroji
Invalidation	předčasné zneplatnění obsahu v keši

Co a kde kešujeme

Strategie pro kešování

- jak často se obsah aktualizuje?
- je obsah individualizován?

1. statický a málo se měnící obsah (nejdelší doba)
   - loga, pozadí, ikony
   - soubory s obecným CSS a JS
   - multimediální soubory
   - soubory ke stažení
2. obměňovaný uživatelský obsah (opatrně, můžeme invalidovat)
   - HTML soubory (třeba s články)
   - měnící se obrázky
   - často upravované CSS a JS
3. obsah nevhodný pro kešování
   - individuální obsah pro přihlášeného uživatele
   - citlivé informace (bankovnictví, webmail…)

ROOT.CZ vpsFree.cz

Kde všude se kešuje

- v uživatelově prohlížeči
  - invalidace pomocí ⇧ + F5 nebo ⇧ + ctrl + R
  - vyprázdní keš a pošle `Cache-Control: no-cache`
- v mezilehlých keších
  - provozovaných mimo vaši infrastrukturu
  - nemáte je přímo pod kontrolou
- reverzní proxy server
  - server v naší infrastruktuře
  - může být součástí webserveru, samostatně nebo CDN

ROOT.CZ vpsFree.cz

HTTP hlavičky

Hlavičky

- kešovací HTTP hlavičky může posílat klient i server
  - proxy vystupuje v obou rolích
- obě strany se informují o svých konfiguracích
- kešující strana může mít svou politiku
  - může invalidovat dříve, ale nikdy **ne později**
- hlavní slovo má obvykle zdroj obsahu
  - jeho konfigurace je *vysílána* právě pomocí hlaviček
- hlavičky **nejsou** citlivé na velikost písmen

Jak si zobrazit HTTP hlavičky

Příkazem Curl

```
$ curl -I https://www.petrkrcmar.cz/
```

Příkazem wget

```
$ wget -S --spider https://www.petrkrcmar.cz/
```

Funkce jednotlivých HTTP hlaviček

Expires	určuje čas v budoucnosti, do kterého je vydávaný obsah platný; poté už se musí klient znovu zeptat
Cache-Control	modernější a flexibilnější varianta téhož; je možné nastavit obě najednou
Etag	unikátní označení obsahu; keš se může kdykoliv zdroje zeptat na jeho platnost
Last-Modified	určuje čas poslední změny; hodí se při některých časově závislých strategiích kešování
Content-Length	velikost přenášeného obsahu; některé keše ukládají jen obsah s předem známou velikosti
Vary	určuje další hlavičky, které zahrnuje keš do svého rozhodovacího klíče

Hlavička Expires

- starší standard překonaný hlavičkou `Cache-Control`
- obsahuje datum a čas expirace daného objektu
  - typ HTTP-date v RFC 7231
- příjemce musí zpracovat i nevalidní datum včetně data v minulosti
- pokud je v odpovědi hlavička `Cache-Control`, příjemce musí `Expires` ignorovat
- hodnota `0` znamená, že platnost záznamu už vypršela

Příklad hlavičky Expires

```
expires: Sun, 06 Oct 2019 20:00:00 GMT
```

Hlavička Cache-Control (1/2)

- umožňuje definovat různé strategie pro kešování vašeho obsahu
- lze poslat více různých instrukcí oddělených čárkou

max-age	čas v sekundách udávající dobu čerstvosti obsahu; maximum je jeden rok
no-cache	kešovaný obsah musí být ihned revalidován; lze použít různé techniky pro validaci ve zdroji
no-store	obsah se nesmí vůbec ukládat; hodí se pro citlivá data
private	obsah může být kešován v prohlížeči, ale ne v mezilehlých keších
public	obsah lze kešovat i v mezilehlých; přebíjí `private`

ROOT.CZ vpsFree.cz

Hlavička Cache-Control (2/2)

s-maxage	stejný jako `max-age`, ale týká se jen mezilehlých keší
must-revalidate	striktní přístup k času vypršení; starý obsah neposílat ani při přerušení komunikace se zdrojem
proxy-revalidate	jako `must-revalidate`, ale jen na mezilehlé; prohlížeč stále může pro případ výpadku kešovat
no-transform	keše nesmí měnit obsah kvůli výkonu; například ho nesmí samy komprimovat

Příklad hlavičky Cache-Control

```
cache-control: public,max-age=3600
```

Hlavička Etag

- umožňuje označit obsah jednoznačným identifikátorem
- server pak nemusí posílat celý obsah pro porovnání
- při změně obsahu se **musí** změnit Etag
  - prostým porovnáním je možné detekovat změnu

Příklad hlavičky Etag

```
etag: c105364f1a847c07860ad7bd9d23eef0
```

Hlavička Last-Modified

- obsahuje časový údaj poslední změny
- čas podle serveru, kdy došlo k poslední změně
- méně přesná než `Etag`, používá se jako záložní
- při použití požadavkových hlaviček `If-Unmodified-Since` a `If-Modified-Since`
  - server vrátí `304`, pokud nedošlo ke změně

Příklad hlavičky Last-Modified

```
last-modified: Sun, 06 Oct 2019 8:00:00 GMT
```

Hlavička Content-Length

- informuje o velikosti přenášeného obsahu v bytech
- některé keše položku vyžadují
- lze podle ní rozhodnout, který obsah nebude kešován

Příklad hlavičky Content-Length

```
content-length: 32768
```

Hlavička Vary

- keš obvykle porovnává dotazy jen podle URL
- hlavička umožňuje rozšířit klíč o další položky v **požadavku**
- například říká, že roli hraje i `User-Agent`
  - když chceme rozlišit mobilní a desktopové klienty
  - nebo různé jazykové verze

Příklad hlavičky Vary

```
vary: user-agent,accept-language
```

Strategie kešování

Než začnete kešovat

- různým součástem webu můžeme nastavit různá pravidla
- různým uživatelům můžeme nastavit různá pravidla
- rozhodněte se, které objekty je možné kdy kešovat
  - dlouhodobě
  - krátkodobě s možností invalidace
  - vůbec
- pozor na unikátní řetězce v URL
  - vedou k unikátním dotazům = keš nefunguje

Obecná doporučení

- oddělte v URL dynamický obsah od statického
  - obrázky, statické soubory, CSS, JS...
- keš máte pod kontrolou jen na svých serverech
  - zvažte stažení externích objektů k sobě
- kešované objekty vyžadující aktualizaci pojmenuje verzí
  - místo `jquery.js` raději `jquery-1.2.3.js`
- používejte jednotné cesty k objektům na všech podstránkách
  - keše se řídí cestou, neměňte ji na různých místech

Volba hlaviček pro různý typ obsahu

- všechny stupně keše by měly držet **statický obsah**
- nastavte dlouhou dobu kešování **podpůrným prvkům**
  - CSS, JS, loga, ikony, fonty…
  - lze je kdykoliv vyměnit, pokud mají unikátní název
- prohlížeče mohou mít právo kešovat **uživatelský obsah**
- bez kešování může zůstat **důležitý proměnlivý obsah** (košík)
- umožněte **validaci bez stahování** a zbytečné zátěže
  - použitím hlaviček `Etag` nebo `Last-Modified`
- nadřazený obsah (HTML) se musí načítat častěji než podřazení
  - jinak neovlivníme například výměnu aktualizovaných objektů

Nginx jako kešující reverzní proxy

Nginx nejen jako web server

- Nginx je web server se spoustou možností
- umí fungovat i jako reverzní proxy
- má velmi mocné kešování
  - v roli web serveru (FastCGI keš)
  - v roli reverzní proxy (proxy keš)
- ve výchozím stavu se kešují jen požadavky `GET` a `HEAD`
- respektuje se hlavička `Cache-Control`
- nekešují se `Private`, `No-Cache`, `No-Store` a `Set-Cookie`
- dle klíče vytváří MD5 pro název souboru

Konfigurace kešující proxy

- pro konfiguraci jsou postačí dvě volby

`proxy_cache_path` nastavuje parametry keše: cestu, klíč...

`proxy_cache` vybírá kešovací zónu pro danou část serveru

Příklad konfigurace

```
proxy_cache_path /path/to/cache levels=1:2
                 keys_zone=my_cache:10m max_size=10g
                 inactive=60m use_temp_path=off;

server {
    # ...
    location / {
        proxy_cache my_cache;
        proxy_pass http://my_upstream;
    }
}
```

Vysvětlení položek

levels	víceúrovňová struktura; snižuje počet souborů v adresáři
keys_zone	pro rychlé hledání ukládá klíče v RAM (1 MB = 8K záznamů)
max_size	maximální velikost diskové keše; nepoužívané odstraňuje
inactive	doba pro odstranění záznamu; nemaže se podle hlaviček
use_temp_path	uloží obsah na stejné místo a pak přejmenuje
proxy_cache	zapne použití keše; může být v http, server, location

Definice klíče

- je možné definovat, jak se sestavuje klíč
  - z interních proměnných poskytovaných serverem
- ze zvoleného řetězce vznikne MD5 pro název souboru
- může se používat vícestupňová (až 3) adresářová struktura
  - pro snížení počtu souboru v adresáři
  - hierarchie má 1 až 3 stupně, počet zvolených znaků je 1 nebo 2

Příklad cesty k souboru

```
/data/nginx/cache/c/29/b7f54b2df7773722d382f4809d65029c
```

Příklad klíče

```
proxy_cache_key "$scheme$proxy_host$request_uri";
```

Čistění keše

- Nginx má mechanismus pro odstranění stránky z keše
- při zavolání správné cesty označí za neplatné
- WordPress má plugin nginx-helper, který sám zavolá

Příklad konfigurace

```
server {
…
location ~ /purge(/.*) {
   allow 127.0.0.1;
   proxy_cache_purge my_cache "$scheme$proxy_host$1";
}
```

Informace v hlavičce

- Nginx umí přidat informace o úspěchu keše
- nachází se v proměnné `upstream_cache_status`
- existuje řada dalších souvisejících proměnných

Příklad konfigurace

```
add_header X-Cache $upstream_cache_status;
```

Význam jednotlivých výsledků

MISS	odpověď nebyla v keši, byla stažena ze zdroje
BYPASS	nebyla použita keš kvůli podmínkám v `proxy_cache_bypass`
EXPIRED	obsah v keši vypršel; odpověď je ze zdroje
STALE	zdroj neodpovídá, odpověď je z keše podle `proxy_cache_use_stale`
UPDATING	odpověď je z keše, nová verze se stahuje
REVALIDATED	keš ověřila validnost obsahu podle `proxy_cache_revalidate`
HIT	v keši byla validní odpověď, kterou jste právě dostali

Děkuji za pozornost

Otázky?

Petr Krčmář
petr.krcmar@iinfo.cz

ROOT.CZ vpsFree.cz