Využitie technológie CSS SPRITE pri tvorbe webových aplikácií

Bc. Roman Mátyus

Obsah

1 Úvod

Technológia CSS Sprite je určená pre zrýchlenie načítavania obrázkov vo webovej aplikácií. Podstatou prínosu je minimalizovanie počtu HTTP požiadaviek na server a zníženie objemu prenášaných hlavičiek obrázkov. Toto sa zabezpečí zľúčením čo najväčšieho počtu obrázkov do jediného zloženého obrázku.

2 Popis technológie CSS Sprite

Pri používaní tejto techniky sú všetky obrázky použité v grafike, ak spĺňajú kritériá, zľúčené tak, aby z nich vznikol jeden obrázok obsahujúci obsah všetkých čiastkových obrázkov.

Následne je pozmenený obsah CSS súbora. Je potrebné identifikovať súradnice a rozmery jednotlivých čiastkových obrázkov. Tieto súradnice sa najčastejšie používajú na vytvorenie nových tried v CSS a sú zapísané v atribútoch `background-position`. Týmito triedami sú následne označené jednotlivé HTML elementy ktorých pozadie má obsahovať daný čiastkový obrázok. Informácie o súradniciach je možné použiť aj priamo na úpravu pôvodného CSS súboru. Takáto úprava je pracnejšia, ale nie je nutné modifikovať pôvodný HTML kód.

2.1 Výstupný súbor PNG

Najčastejším výstupným formátom pri tejto technike je PNG.

Tento súborový formát v sebe zľučuje viacero výhod:

- relatívne malá výsledná veľkosť,
- bezstratová kompresia,
- priehľadnosť pozadia.

Pri jeho použití treba dbať aj na jeho dve obmedzenia:

- nepodporuje animácie ako GIF,
- Internet Explorer 6 (a staršie) nepodporujú priehľadnosť.

Použitie PNG nemusí byť vyslovene pravidlom, pretože vhodne vytvorený GIF súbor, resp. JPG súbor s kompresiou môžu dosahovať lepšie výsledky. Všetko závisí od vstupných súborov. Ak by bolo na vstupe napríklad viacero väčších obrázkov v truecolor, tak by malo zmysel uvažovať o JPG výstupe. JPG bude úplne diskvalifikovaný, ak bude niektorý obrázok na vstupe obsahovať priesvitnú časť.

2.2 Výhody

2.2.1 Menej HTTP požiadaviek

Namiesto množstva samostatných požiadaviek server vráti jediný obrázok s jednou hlavičkou. Vzhľadom na dlhšie komunikačné časy na mobilných sieťach a kvôli stále väčšiemu podielu mobilných zariadení na prístupoch na webové aplikácie je toto riešenie veľkým prínosom.

Pri požiadavkách na server sa prenáša relatívne mnoho informácií[1]. Tieto informácie sú teda v bežnom prípade prenášané pre každý obrázok samostatne a teda dochádza k mnohonásobnej duplicite prenosu.

[1] Hlavičky obsahujúce informácie o prehliadači, cookies a iné.

2.2.2 Úspora prenášaných dát

Každý prenášaný obrázok obsahuje okrem obsahovej časti aj svoju hlavičku. V prípade spojenia obrázkov do jedného, je ušetrený prenos hlavičiek pre každý súbor zvlášť.

Táto vlastnosť nemusí byť vždy výhodou. Často sa stáva, že obrázky nie je možné vo výslednom obrázku optimálne rozmiestniť a výsledný obrázok má väčšiu plochu (px^2) ako je súčet plôch samostatnch obrázkov. Vďaka kompresií obrázka to nemusí nutne znamenať priamo úmerné zväčšenie dátového súboru obrázka.

Sprite je možné po vygenerovaní optimalizovať.

2.2.3 Okamžitá dostupnosť obrázka

Webový grafici často vyžadujú zmenu grafiky po interakcií kurzora[^interakcie kurzora] s nejakým elementom na stránke. Toto je zväčša prakticky vyriešené zmenou obrázka pozadia na úrovni CSS. V tomto prípade nastáva problém s oneskoreným načítaním pozadia. Prehliadač zaregistruje potrebu vykreslenia obrázka, ale ten ešte nie je k dispozícií. Preto oň server požiada a stiahne ho. Toto môže bežne trvať približne pol sekundy. Keďže toto prebliknutie kazí používateľský zážitok z používania aplikície, dizajnéri niekedy vytvárajú čiastkové *sprity* zložené z pôvodného pozadia a pozadia po interakcií. Toto je žiaľ pracné a nesystémové riešenie.

[^interakcie kurzora]: Umiestnenie kurzora nad element, kliknutie naň a iné.

2.3 Nevýhody

2.3.1 Pracnosť

Pri manuálnom vytvorení *spritu* je potrebné spojiť všetky vyhovujúce obrázky do jedného, vyrátať pozície jednotlivých obrázkov a vypočítané pozície zápisať do CSS súboru. Najpracnejším spôsobom je spracovávať požadované súbory v grafickom editore a manuálne dopisovať potrebné pozície do CSS súboru.

Možným zjednodušením je importovanie zdrojových súborov do nástroja generujúceho *sprite* obrázok použitím vhodného algoritmu. Takto získaný súbor je potrebné manuálne spracovať – aplikovať pozície zdrojových obrázkov do požadovaného CSS súboru.

Niektoré generátory poskytujú ako výstup automaticky triedy CSS štýlov s pozíciami pôvodných obrázkov na základe vygenerovaného obrázka. Tieto triedy sa dajú následne jednoducho aplikovať priamo v HTML prípadne sa dajú použiť CSS preprocesorom SASS, LESS alebo inými.

2.3.2 Náročnosť generovania

V prípade použitia generátora za behu aplikácie by mohlo vzniknúť oneskorenie z dôvodu výpočtovej náročnosti generovania obrázka a jeho následné dosadenie do zdrojového súboru. Keďže žiaľ funkčné

nástroje generujúce *CSS Sprite* za behu aplikácie aktuálne neexistujú, nie je to akútny problém. Problém je taktiež jednoducho riešiteľný kešovaním.

2.3.3 Veľkosť obrázka

V ideálnom prípade by sprite obrázok obsahoval množinu obrázkov zhodných rozmerov. Výsledný obrázok by teda mal plochu v px^2 zhodnú so súčtom plôch jednotlivých obrázkov. V reálnom nasadení sa toto nestáva a teda pri tvorbe obrázka dochádza k vzniku prázdnych miest. Tento efekt je síce nežiadúci, ale nie kritický, pretože prázdne miesta zaberajú relatívne málo bytov.

Dátová veľkosť výsledného obrázka je často dokonca menšia ako súčet veľkostí vstupných obrázkov. Ak sú vstupné obrázky vhodných rozmerov nevzniká nevyplnený priestor a ušetrí sa prenos n-1 hlavičiek súborov. Úspora vzniká aj vyššou mierou kompresie väčšieho obrázka oproti kompresii malých vtupných obrázkov.

Problém prázdneho miesta v *sprite* je čiastočne riešiteľný:

- použitím vhodného algoritmu rozvrhnutia elementov,
- zvolením najvhodnejšieho výstupného súboru prípadne
- použitia kompresie pri ktorej nedochádza k evidentnému znehodnoteniu kvality výstupu.

Vhodné podmienky výstupu je možné pri niektorých aktuálne použiteľných generátoroch manuálne nastaviť. Jednoduchým riešením je aj zmenšiť farebnú hĺbku výstupu.

Zvoliť vhodný algoritmus volieb výstupného súboru nemusí byť vždy ľahké. Hlavne v prípade ak sú na vstupe výborne optimalizované obrázky, môže byť výstup zo štandardných generátorov bez extra nastavenia, prípadne ďalšej úpravy v externom editore, výrazne dátovo väčší ako súčet veľkosti dát obrázkov na vstupe. V takomto prípade treba previesť merania a zistiť či sa oplatí takto pregenerovaný súbor nasadiť.

Vždy to záleží od konkrétneho nasadenia, je však veľmi pravdepodobné, že sa aplikácia techniky napriek väčšiemu objemu oplatí vďaka úspore HTTP požidaviek.

2.3.4 Zväčšenie obsahu CSS

Použitím CSS Sprite je nutné preniesť informácie o pozicovaní obrázkov k používateľom a teda zväčšiť množstvo prenášaných dát. Generátory zväčša vytvárajú vlastné sekcie do CSS súborov ktoré sa majú aplikovať priamo v HTML kóde. Dopad tejto nevýhody je možné minimalizovať napríklad tak, že namiesto dopísania nových CSS štýlov sú dodané štýly pozmenené a tak následne distribuované. Toto sa v praxi používa v závislosti od schopností generátora prípadne kódera ktorý pracúva CSS Sprite obrázok manuálne. Väčšina generátorov touto funkcionalitou nedisponuje vzhľadom na relatívne zložitú implementáciu keďže je mnoho spôsobov ako môže dôjsť ku sporom v definícií štýlov a následnému znehodnoteniu dizajnu.

2.3.5 Úprava HTML

Veľkou nevýhodou je pri použití štandardných generátorov nutnosť meniť CSS triedy súborov priamo v HTML kóde webu. Tieto úpravy môžu byť niekedy pracné a neprehľadné. Podstatne vhodnejšie by bolo iba upraviť kód v triedach metódou sspomenutou v predošlom odstavci. Tým by bola zachovaná spätná kompatibilita.

2.3.6 Obmedzenia CSS

Samotný návrh štandardu CSS neobsahuje dokonalú podporu pre CSS Sprite. Táto technika je preto použiteľná iba na určitú množinu obrázkov, nie na všetky.

Vzorka Štandardne nie je možné túto techniku využiť na obrázky ktoré sú vzorkou - opakovane vypĺňajú pozadie v nejakom smere.

V prípade ak sa to oplatí, je možné toto obmedzenie čiastočne eliminovať a vygenerovať viac typov obrázkov pre každý typ opakovania. Jeden *sprite* pre obrázky ktoré sa neopakujú a ďalšie zvlášť pre horizontálne a vertikálne opakované pozadia. Dôsledkom môžu byť až tri výsledné súbory. Ak je na stránke definovaných väčšie množstvo horizontálne/vertikálne sa opakujúcich pozadí môžu sa tým získať výrazné úspory.

Z technológie vyplýva, že v prípade horizontálneho opakovania by musel byť každý obrázok najprv upravený na šírku toho najširšieho. Obdobne to platí pre vertikálne opakované obrázky. Obrázky ktoré sa podla definície majú opakovať v horizontálnom aj vertikálnom smere nemôžu byť v CSS Sprite obrázkoch vôbec použité.

Pozicovanie pozadia Problém nastáva aj v prípade, ak je pozadie elementu pozicované relatívne.[2] Pozície musia byť buď nulové, alebo určené v pixeloch.

2.3.7 Formát vstupných súborov

Z dôvodu nepodporovania animácie pri PNG súboroch[3], nemôže byť štandardne do *spritu* zaradený žiadny animovaný GIF. Toto obmedzenie sa dá teoreticky obísť vďaka simulácií animácie použitím techník CSS3.

2.4 Konkurenčné riešenie Data URL

Pre úplnosť je vhodné spomenúť techniku urýchlenia webu pomocou Data URL[4]. Pri tejto technike je v CSS odkaz na súbor (obrázok) nahradený za hlavičku a textovú reprezentáciu obrázka kódovaním base64. Hlavnou nevýhodou je zväčšenie prenášaného množstva dát o 33% oproti originálu.

[2] Percentuálne alebo top, bottom, left, right a center.

[3] Formát PNG je štandardne používaný ako výstupný formát *sprite* obrázka.

[4] http://tools.ietf.org/html/rfc2397

Možným obmedzením je problematická podpora pseudoprotokolu `data` v Internet Exploreri 8 a starších. IE 8 podporuje *Data URL* s maximálnou veľkosťou 32 kB a staršie nepodporujú túto funkciu vôbec.

	štandardné volanie	Data URI	CSS Sprite
Počet HTTP komunikácií pre zobrazenie n obrázkov	n	0	1
Veľkosť dát oproti originálu	zhodná	+ 33 %	± zhodná
Cache	možné	nutnosť cachovania CSS	možné

Tabuľka 1: Porovnanie štandardného volania obrázkov, Data URI a CSS Sprite

Štruktúra takéhoto reťazca je podla špecifikácie nasledovná:

```
data:[<mediatype>][;base64],<data>
```

2.4.1 Príklad

Namiesto

```
background: white url('dot.png')
```

je cesta k súboru v CSS nahradená za textovú reprezentáciu obrázka a štýl vyzerá nasledovne:

```
background: white url('
EUgAAAAUAAAAFCAYAAACNbyblAAAAHElEQVQI12P4//8/w38GIAXDIBKE0DHxglj
NBAAO9TXL0Y4OHwAAAABJRU5ErkJggg==')
```

3 Referenčný dizajn

Ako referenčný dizajn pre testovanie postupov, rôzne merania a analýzu som pri práci zvolil open source šablónu pre CMS Joomla s názvom Eclipse[5]. Šablóna bol vybratá náhodne. Podmienkou výberu bol moderný dizajn a otvorený zdrojový kód pre licenčne bezproblémové použitie v práci. Šablónu je možné stiahnuť z webstránky joomlatemplates.me.

Obrázok 1: Šablóna Eclipse

3.1 Analýza podkladov

Šablóna obsahuje spolu 6 CSS súborov. Súbory `bootstrap.min.css`, `font-awesome.min.css`, `font-awesome-ie7.min.` neobsahujú žiadne obrázky, takže nie sú relevantné pre spracovanie. Súbor `flexslider.css` obsahuje iba jediný obrázok, takže *sprite* taktiež nemá zmyseľ.

V súbore `style.css` sa nachádza definícia vzhľadu samotnej šablóny.

Obsahuje 17 obrázkov s nasledujúcimi definíciami:

Názov súboru	Použiteľný pre CSS Sprite?	Veľkosť	Rozmer	Plocha

[5]http://joomlathemes.co/demo3x/eclipse

bg.png	horizontálne	19,0 kB	50 x 600 px	30 000 px²
header.png	áno	18,8 kB	980 x 600 px	588 000 px²
facebook.png	áno	1,9 kB	32 x 32 px	1 024 px²
twitter.png	áno	2,1 kB	32 x 32 px	1 024 px²
google.png	áno	2,0 kB	32 x 32 px	1 024 px²
youtube.png	áno	2,0 kB	32 x 32 px	1 024 px²
dribbble.png	áno	2,8 kB	32 x 32 px	1 024 px²
flickr.png	áno	2,0 kB	32 x 32 px	1 024 px²
pinterest.png	áno	2,2 kB	32 x 32 px	1 024 px²
picasa.png	áno	2,6 kB	32 x 32 px	1 024 px²
linkedin.png	áno	1,9 kB	32 x 32 px	1 024 px²
reddit.png	áno	1,9 kB	32 x 32 px	1 024 px²
more.png	horizontálne	1,4 kB	1 x 36 px	36 px²
sidebar-li2.png	nie	1,5 kB	7 x 7 px	49 px²
sidebar-li1.png	nie	1,5 kB	7 x 7 px	49 px²
arrow_up.png	nie	1,3 kB	50 x 50 px	2 500 px²
error.png	áno	1,9 kB	32 x 32 px	1 024 px²
SPOLU		**62,9 kB**		**631 898 px²**

Tabuľka 2: Detailný rozbor obrázkov

Celková veľkosť súborov v šablóne je 62,9 kB.

4 Manuálne vytvorenie CSS Spritu

V programe Gimp[6] som čo najoptimálnejšie manuálne vyskladal obrázky. V

29 kB 42,2 raw po jednom

4.1 Výpočet užitočnosti

4.1.1 Objem dát

4.1.2 Šetrenie HTTP požiadaviek

Vplyv optimálnosti zloženia výsledného obrázky na veľkosť PNG.

[6] http://www.gimp.org

5 Existujúce generátory

5.1 Sprite Me

http://spriteme.org/

5.2 CSS Sprite Generator (Project Fondue)

http://spritegen.website-performance.org/

5.3 CSS Sprites Generator

http://csssprites.com/

5.4 CSS Sprite Creator

http://www.web2generators.com/graphism/css_sprite_creator

6 Vlastné riešenie

6.1 Popis funkčnosti

6.2 Riešenie netriviálnych problémov

6.2.1 Opakovanie

Ak chceme do *spritu* zaradiť aj obrázok ktorý je vykraslovaný ako opakované pozadie, musí sa v danej osi nachádzať v celej šírke/výške obrázka.

6.2.2 Obmedzenie CSS

6.2.3 Formát súborov

6.3 Spätná väzba

7 Záver