UX + pagespeed
Quick scan

bedrijfslogo



27 MAART

ERWIN HOFMAN
Erwinhofman.nl

Introductie

Gezien toename van mobiel verkeer, zijn in dit document de mobiele testresultaten steeds het uitgangspunt.

Binnen trajecten is PageSpeed Insights een snel te gebruiken middel om de hygiëne in kaart te brengen, deze is immers door iedereen te benaderen. Aanvullend gebruik ik ook video-metingen, om echte gebruikerservaring er bij te betrekken.

PageSpeed Insights:
https://developers.google.com/speed/pagespeed/insights/

Belangrijke PageSpeed Insights metric

Alhoewel de algehele score een goede samenvatting en indicatie van de performance hygiëne is, adviseer ik altijd te kijken naar de individuele metrics. Deze brengen namelijk de impact op de gebruikerservaring accurater in kaart.

De twee metrics in het bijzonder:

- First Meaningful Paint (FMP)
  Hoe snel ziet de bezoeker betekenissvolle content. Denk in het geval van een product detailpagina aan de titel, prijs, foto.
  Hiermee heb je ook de TTFB en FCP metrics te pakken.

- Time to Interactive (TTI)
  Geeft aan op welk punt in de tijdlijn de browser klaar is met afhandelen van door de webbouwer/platform opgelegde taken. Een slechte TTI metric kan resulteren in slechte gebruikerservaring, doordat de interactie met de website nog niet of niet volledig aangegaan kan worden.
  Dit kan het idee van slecht of niet functionerende user-flow in de hand werken, met uitstap/bounce als gevolg.

Waar FMP vervangen gaat worden door LCP, krijgt TTI binnen een Lighthouse test een kleiner gewicht. Total Blocking Time (TBT) komt er voor in de plaats.
https://www.erwinhofman.nl/blog/performance-metrics-lcp-replacing-fmp/

Performance bottleneck voor smartphones

TTI ondervangt ook direct andere JavaScript gerelateerde metrics, waaronder de First Input Delay. Weet je de TTI te reduceren, dan zal dit ook positieve uitwerking hebben op First Input Delay, First CPU Idle en Total Blocking Time.

De reden dat vanuit PageSpeed Insights en Lighthouse op deze metrics ligt, is vanwege de CPU boundaries binnen smartphones. Slechte metrics vormen een performance en daarmee UX bottleneck, waar voorheen hoeveelheid data vooral een bottleneck vormde.

Deze verschuiving heeft alles te maken met de verandering in hoe we websites zijn gaan bouwen. Deze werden groter, uiteindelijk moest het responsive ten behoeve van smartphones. Ondertussen zijn we als developers smartphones dezelfde huiswerk mee gaan geven, als desktop apparaten. Eén van de redenen dat websites nog steeds langzamer worden.

Ook wordt er tegenwoordig nog vooral naar de hosting gewezen bij langzame websites, waar de voorkant en dus het werk van een frontend ontwikkelaar reeds minstens zo belangrijk is geworden.
Deze rol was voorheen minder groot, waardoor frontend performance kennis geen gemeengoed was.

Daarbij worden door frontend ontwikkelaars soms gegrepen naar standaard oplossingen, zoals plugins, in de poging de score te verbeteren. Uitkomsten kunnen zijn dat scores slechts licht verbeteren, de verbeteringen niet reflecteren naar echte gebruikers of het in het ergste geval zelfs anti-patterns oplevert, waar het binnen het vorige HTTP tijdperk (HTTP/1) nog tot best practices behoren.

Page-type PSI resultaten

Home (61%)

First Contentful Paint	1.7 s	First Meaningful Paint	1.7 s
Speed Index	4.1 s	First CPU Idle	9.1 s
Time to Interactive	9.3 s	Max Potential First Input Delay	750 ms

Categorie (15%)

First Contentful Paint	6.4 s	First Meaningful Paint	7.6 s
Speed Index	13.1 s	First CPU Idle	7.6 s
Time to Interactive	11.8 s	Max Potential First Input Delay	220 ms

Product detailpagina (44%)

First Contentful Paint	1.7 s	First Meaningful Paint	1.7 s
Speed Index	8.6 s	First CPU Idle	8.8 s
Time to Interactive	9.7 s	Max Potential First Input Delay	250 ms

Overeenkomsten

Er zijn enkele duidelijke overeenkomsten in de resultaten.

- De FMP metrics liggen voor Home en DPD in het groen;
- De TTI metric ligt consequent in het rood en levert slechte performance en UX;
- De speed Index komt laat op gang.

Omtrent de FMP is op aanbevolen wijze een best practice toegepast voor het inladen van externe stylesheets in cominatie met critical CSS. Daarmee zijn echter anti-patterns geïntroduceerd.

Anti-patterns

Bij het optimaliseren van snelheid, kunnen anti-patterns onstaan: bottlenecks die op snelle verbindingen niet ervaren worden, maar waar de ervaring op slechtere verbindingen, oudere telefoons of minder up-to-date software kan verslechteren.

CLS metric

Impact op performance, of in dit geval UX, is terug te zien in het laadproces van de homepagina:



Men ziet eerst een half opgemaakte pagina, waarbij de tekst is opgemaakt, maar niet correct gepositioneerd is. Iemand kan vanwege verschillende factoren die je als ontwikkelaar niet in de hand hebt, te lang naar een dergelijke tussenstand kijken.

Ook op snellere verbindingen krijgt men een schokkerige ervaring voorgeschoteld. Hier is een metric voor, namelijk CLS. Dit staat voor Cumulative Layout Shift, en brengt de procentuele layout verschuivingen tijdens het laadproces in kaart. CLS wordt in ieder geval onderdeel van Lighthouse v6.

Verschuivingen binnen het laadproces op een een desktop met snelle 3G, is hier goed te zien:
[anonimyzed webpagetest video url]

Het CLS percentage ligt op 92,92%.

Resolution	{"absolute":{"height":1200,"width":1920},"available":{"height":1200,"width":1920}}
Colordepth	24
CumulativeLayoutShift	0.9292119916
Dpi	{"dppx":1,"dpcm":37.79527559055118,"dpi":96}

Idealiter zou deze echter rond de 10% moeten liggen, om juist gebruiksfrustratie te voorkomen. In onderstaand voorbeeld van een andere shop bedraagt de CLS 6,78%

Colordepth	24
CumulativeLayoutShift	0.0678355543
Dpi	{"dppx":1,"dpcm":37.79527559055118,"dpi":96}

Critical CSS

Inlinen van CSS, heeft twee gevolgen:

- Je verliest het voordeel van client side caching, waardoor on-site ervaring verslechtert;
- Je document-size neemt toe, als ook je content-download-time. Dat betekent dat het effect op langzamere telefoons minder optimaal zal zijn, waar op snellere toestellen het effect vanwege snelheid al minimaal was;
- Indien Critical CSS on-the-fly wordt toegepast, kan dit zelfs voor impact op de Time to First Byte zorgen.

Dit laatste zit als volgt:
De homepagina met critical CSS bedraagt 224kb, zonder critical CSS 144kb (uncompressed). Dat betekent een content-size toename van 55%.

Time to First Byte

Hoe snel de browser de geleverde code kan renderen en aan het werk kan, hangt af van de snelheid van zowel hosting en (meerendeels) platform. De TTFB van de homepagina lijkt consequent in goede staat.

De TTFB metric voor respectievelijk een categorie en product detailpagina vallen standaard in het rood.



De geadviseerde grens bedraagt 0.2 seconden. Critical CSS techniek kan vooral profiteren wanneer ook de serverside laadtijden optimaal zijn.

Resource hints

Resource-hints kunnen praktisch zijn, maar weerleggen binnen browsers de prioriteiten. Binnen de website zijn resource hints ongewenst ingezet, waardoor dit conflicteert met de werking en prioritering van de browser zelf.

Minification en client side caching

Er is sprake van minification, bijvoorbeeld binnen stylesheets. Gevolg is eenzelfde euvel dat ook in Magento 1 onopgelost bleef: het voordeel van client side caching vervalt. Elke type pagina lijkt zijn eigen stylesheet-filename te hebben, terwijl stylesheets grotendeels dezelfde inhoud hebben. Dit vormt een on-site bottleneck en staat performance in de weg.

Ongebruikte data

Elke site laadt data in dat door een website uiteindelijk niet gebruikt wordt. Soms 3e graads (denk aan Facebook, Pinterest plugins), soms 1e graads.

Binnen de huidige website is hier ook sprake van, waarbij de percentages ongebruikte kilobytes vrij hoog ligt. Ik beperk me hierbij bij de functionele bronnen (oftewel, exclusief tracking bronnen).

URL	Type	Total Bytes	Unused Bytes	Usage Visualization
/8743de0984296fa1f4c478140ca00e71.css	CSS	1 352 981	1 274 744 94.2 %	
/e69dca1609c416b92de75d7b4ac1c109.js	JS (per...	816 365	389 684 47.7 %	
/css?family=Roboto%3A100%2C100italic%2C200%2C200italic%2	CSS	77 114	77 114 100.0 %	
https://	CSS+J...	90 788	57 647 63.5 %	
https:// /43c3ceae84c31db8e8f86473b36.js	JS (per...	76 980	38 784 50.4 %	
https:// 7d36e28d19710b882a6824e213.js	JS (per...	45 229	13 156 29.1 %	
https:// wp-content/plugi.../lazyload.min.js	JS (per...	5 585	1 382 24.7 %	
https:// om/plugins/ua/ec.js	JS (per...	2 779	1 199 43.1 %	

Maar liefst 1,2MB aan CSS dat gedownload èn geparsed werd, wordt niet gebruikt (94.2%). 0,37MB JS wordt niet gebruikt (47.7%). Dit zijn performance bottlenecks.

Conclusie

De voornamelijk te trekken conclusie is dat er veel focus is uitgegaan naar critical CSS en lazyloading, om in ieder geval de twee gerelateerde PageSpeed Insights recommendations (waaronder de welbekende ‘render blocking resources’) te kunnen repareren.

Beide is gelukt voor de Homepagina en DPD, getuige een PageSpeed Insights audit. Echter, hierbij is vooral gekeken naar de score, in plaats van de impact op de werkelijke gebruikerservaring, en is bovendien voorbij gegaan aan hoe browsers exact werken om tot een goede gebruikerservaring te komen.

Met andere woorden: op papier is er winst behaald, waar de uitwerking in praktijk niet beter is geworden en vanuit een andere UX invalshoek tot verhoogde bounce kan leiden.
Dit is ook in de gedane webpagetest analyse onder ‘CLS metric’ als video te zien.

Vervolg

Binnen de huidige stand van zaken is het naast her-evaluatie en herschrijven van critical CSS ook zaak dat er aandacht uit gaat naar JavaScript, interpretatie van de code door browsers (resource hints, client side caching) en vervolgens werkelijke gebruikerservaring.