Universitatea Națională de Știință și Tehnologie POLITEHNICA București
Facultatea de Automatică și Calculatoare

Studiul tehnicilor de Lazy Loading in aplicatiile web

Materia: *Stiluri arhitecturale și șabloane de proiectare software*

Conducător științific
BOIANGIU Costin Anton

Student
BÂRA Răzvan-Mihail

Anul 2023

Cuprins

Lista figurilor

Capitolul 1
Introducere

În oricare pagină ale unei aplicații web trebuiesc încărcate mai multe resurse. Încărcarea resurselor într-o pagină web implică recuperarea și redarea diferitelor tipuri de conținut pentru a afișa pagina completă utilizatorului. Iată o prezentare generală a modului în care se încarcă diferite tipuri de resurse într-o pagină web:

1. HTML (Hypertext Markup Language):
   - Atunci când utilizatorul introduce o adresă URL sau accesează un link, browserul trimite o cerere către server pentru un document HTML.
   - Serverul răspunde prin trimiterea fișierului HTML, care conține structura și conținutul de bază al paginii web.
2. CSS (Cascading Style Sheets):
   - Odată ce browserul primește documentul HTML, este analizat și identifică orice fișier CSS asociat paginii. Apoi, browserul trimite cereri suplimentare către server pentru a prelua fișierele necesare. CSS este esențial pentru a stiliza conținutul HTML, pentru a defini layout-uri și pentru a îmbunătăți prezentarea vizuală a paginii.
3. JavaScript:
   - Similar cu CSS, fișierele JavaScript asociate paginii HTML sunt identificate.
   - Browserul solicită și încarcă fișierele JavaScript de pe server. JavaScript este responsabil de îmbunătățirea interactivității paginii, de gestionarea interacțiunilor utilizatorului și de actualizarea dinamică a conținutului.
4. Imagini:
   - Imaginile incluse în documentul HTML sunt încărcate de către browser sub formă de cereri separate către server. Formatele de imagine obișnuite includ JPEG, PNG, GIF și SVG. Imaginile de dimensiuni mari pot fi comprimate pentru a reduce timpul de descărcare.

   Fonturi:
   - Dacă în CSS sunt specificate fonturi personalizate, browserul preia aceste fișiere de fonturi de pe server.
   - Fonturile sunt esențiale pentru menținerea unei tipografii coerente pe diferite dispozitive.
5. Alte resurse externe (de exemplu, video, audio):
   - Resurse precum videoclipurile și fișierele audio sunt încărcate ca cereri separate.

- Browserul poate utiliza elemente HTML5 native (de exemplu, ¡video¿, ¡audio¿) sau plugin-uri externe pentru a gestiona conținutul multimedia.

6. Încărcare asincronă:

   - Paginile web moderne folosesc adesea tehnici de încărcare asincrone pentru a accelera încărcarea inițială a paginii. Scripturile și resursele pot fi marcate ca fiind asincrone, permițând browserului să continue analiza și redarea paginii în timp ce, în același timp, preia și execută aceste resurse.

7. Caching:

   - Browserele stochează resursele la nivel local pentru a îmbunătăți performanța la vizitele ulterioare.
   - Resursele stocate în memoria cache sunt recuperate din memoria locală în loc să se facă noi cereri către server, ceea ce reduce timpul de încărcare.

8. Calea critică de redare:

   - Calea critică de redare reprezintă secvența de pași pe care browserul îi parcurge pentru a converti HTML, CSS și JavaScript într-o pagină adecvată prezentării utilizatorului.
   - Optimizarea căii critice de redare este crucială pentru timpi rapizi de încărcare a paginilor.

   Înțelegerea modului în care aceste resurse se încarcă într-o pagină web este esențială pentru optimizarea performanței și pentru a oferi o experiență de utilizare fără întreruperi. Tehnici precum minificarea, compresia și încărcarea asincronă pot fi utilizate pentru a spori eficiența încărcării resurselor.

1.1 Eager loading

În peisajul dinamic al dezvoltării web, urmărirea unei performanțe optime și a unei experiențe de utilizare fără probleme rămâne în topul priorităților al dezvoltării unei aplicații. Pe măsură ce utilizatorii solicită din ce în ce mai mult timpi de încărcare mai rapizi și interacțiuni mai receptive, dezvoltatorii se confruntă cu provocarea de a gestiona eficient recuperarea și încărcarea resurselor pe paginile web. Ca răspuns la această provocare, eager loading, a apărut ca o paradigmă strategică, oferind o abordare a gestionării resurselor cu scopul de a minimiza latența și de a spori satisfacția utilizatorilor.

Eager loading, ca strategie, se seamănă unui concept de anticipare și recuperare a resurselor înainte ca acestea să fie solicitate în mod explicit. Această abordare anticipativă contrastează cu strategiile tradiționale de încărcare a resurselor la cerere, care vizează reducerea întârzierilor asociate cu recuperarea elementelor esențiale în timpul interacțiunilor cu utilizatorii.

Cel mai simplu exemplu de eager loading intr-o pagina web reprezintă solicitarea tuturor imaginilor, scripturilor, foilor de stil și a datelor în momentul încărcării acesteia. Astfel, se elimină comunicarea extensivă cu serverul, toate resursele fiind disponibile la redarea paginii, traficul rețelei este optimizat. Această opțiune este cea implicită în browsere fie că e vorba de framework-urile SPA sau pagini web.

1.2 Lazy loading

În timp ce tehnica eager loading se face remarcată în reducerea numărului de cereri trimise către server, dezvoltatorii trebuie să găsească un echilibru pentru a evita căutarea excesivă și încărcarea inutilă a resurselor. Excesul de căutare poate duce la creșterea utilizării lățimii de bandă și la o potențială scădere a performanței. Prin urmare, este necesară o abordare atentă pentru a optimiza eficiența fără a introduce costuri suplimentare inutile.

Ca o soluție, se dorește încărcarea resurselor numai în momentul în care acestea sunt explicit cerute de utilizator sau documentul HTML determină necesitatea imediată a acestora. Această soluție poartă numele de *Lazy Loading*. De exemplu, imaginile care se afla în josul unei pagini vor fi încărcate când utilizatorul navighează în apropierea acestora. Totuși, nu se limitează doar la imagini, ci se aplică și la scripturi și alte componente. De exemplu, componentele JavaScript sau funcționalitățile necesare pentru anumite interacțiuni cu utilizatorul pot fi încărcate la cerere, sporind eficiența aplicației.

Această tehnică este deosebit de valoroasă în scenariile în care reducerea timpilor de încărcare inițială a paginilor și conservarea lățimii de bandă sunt prioritare. Prin prioritizarea încărcării resurselor critice și amânarea celor neesențiale, utilizatorii pot interacționa mai repede cu conținutul vizibil, creând o experiență pozitivă pentru aceștia.

În timp ce tehnica lazy loading este avantajoasă pentru performanță, dezvoltatorii trebuie să ia în considerare implicațiile sale pentru optimizarea pentru motoarele de căutare (SEO). Motoarele de căutare pot acorda prioritate conținutului care este imediat vizibil, astfel încât dezvoltatorii ar trebui să se asigure că conținutul critic legat de SEO este încărcat în mod convențional.

Capitolul 2
Tema propusă

Acest proiect își propune analizarea diferențelor prin măsurarea performanței și a lațimii de bandă folosită în cadrul aplicațiilor web care folosesc lazy loading sau eager loading. În plus, se dorește compararea SEO-ului aplicațiilor în ambele cazuri. Datele necesare măsurătorilor vor fi preluate cu ajutorul Chrome DevTools și extensiei open-source pentru browser numită Lighthouse.

Pentru realizarea măsurătorilor, se vor dezvolta două aplicații:

- Un website static dedicat aprecierii naturii
- O aplicație full-stack de tip biblioteca online

2.0.1 Website-ul static

Website-ul este dedicat publicării și distribuirilor pozelor cu peisaje interesante din natură. Pozele alese vor fi de o rezoluție mare, într-un număr ridicat. Pagina va avea 5 secțiuni, fiecare va scoate în evidentă o formă de relief. Fiecare dintre acestea, va prezenta conținut sub forma unei scurte descrieri, împreuna cu un slider de 8-10 poze. Astfel, pagina va fi lungă, fiind un exemplu solid de testare și măsurare a performanțelor eager-loading si lazy-loading. Prin aceste exemplu se va evidenția efectul întârzierii încărcării imaginilor asupra timpului total de încărcare a paginii și lățimii de bandă folosite. Pozele vor fi preluate din diferite surse, inclusiv un server local pentru a observa cât de mult sursa imaginilor afectează timpul de încărcare. Proiectul va fi compus dintr-o singura pagină care folosește HTML, CSS și JavaScript.

Față de eager-loading, mă aștept ca tehnica lazy-loading sa prezinte un timp de încărcare mai bun și aceeași lățime de bandă să fie folosită progresiv în urma navigării întregii pagini.