KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Emilis Markulis

GRAFINIŲ ELEMENTŲ UŽKROVIMO TINKLALAPYJE TYRIMAS

Baigiamasis magistro projektas

Vadovas

Doc. dr. Sigitas Drąsutis

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

GRAFINIŲ ELEMENTŲ UŽKROVIMO TINKLALAPYJE TYRIMAS

Baigiamasis magistro projektas

Studijų programos pavadinimas (kodas M4016M21)

Vadovas

(parašas) Doc. dr. Sigitas Drąsutis

(data)

Recenzentas

(parašas)

(data)

Projektą atliko

(parašas) Emilis Markulis

(data)

KAUNAS, 2016

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Emilis Markulis

(Studento vardas, pavardė)

Informatika, M4016M21

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Grafinių elementų užkrovimo tinklalapyje tyrimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 ___ . _____ ____ .

Kaunas

Patvirtinu, kad mano, **Emilio Markulio**, baigiamasis projektas tema „Grafinių elementų užkrovimo tinklalapyje tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

_____ _____

(vardą ir pavardę įrašyti ranka) *(parašas)*

TURINYS

PAVEIKSLĖLIŲ SĄRAŠAS

LENTELIŲ SĄRAŠAS

Emilis Markulis. Grafinių elementų užsikrovimo tinklalapyje tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Sigitas Drąsutis; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatika

Reikšminiai žodžiai: *grafiniai elementai, tinklalapis, ekrano dydis, užsikrovimas, pralaidumas, serveris.*

Kaunas, 2016. 53 p.

SANTRAUKA

XXI amžiaus antrasis dešimtmetis iš svetainių pareikalavo lankstumo, t. y. prisitaikymo prie įvairių interneto vartotojų. Todėl šio tiriamojo darbo tikslas – palyginti grafinių elementų užkrovimo metodus įvairaus dydžio naršyklėse. Tyrime analizuojami keturi skirtingi metodai (HTML5, PHP, COOKIES, RESRC), parenkantys ir užkraunantys reikiamo dydžio paveikslėlius tinklalapiuose. Šie metodai rasti ir suprojektuoti atlikus literatūros analizę. Metodų tyrimas vykdomas populiariausiuose naršyklės ekrano dydžiuose (1920, 1366, 1024, 768, 480px). Vienam metodui buvo atlikti penki bandymai skirtingų dydžių naršyklėse po 50 kartų. Tyrimo rezultatai lyginami pagal šiuos kriterijus: užsikrovimo greitis (s), pralaidumas (KB), kaina ir palaikymas.

Markulis, Emilis. *A RESEARCH ON LOADING GRAPHICAL ELEMENTS FOR WEB PAGES (Each Word of the Title is Written in Capital Letters)*: Master‘s thesis in Informatics / supervisor assoc. dr. Sigitas Drąsutis. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Informatics

Key words: *graphical elements, web page, screen size, loading, bandwidth, server*.

Kaunas, 2016. 53 p.

SUMMARY

In the second decade of the twenty-first century, the websites were ought to become more flexible to conform to different internet users. Therefore, the aim of this research is to compare loading methods (HTML5, PHP, COOKIES, RESRC) of graphical elements in different size web browsers. In this research, four different methods selecting and loading the right size images in the sites were analysed. These methods were found and designed after conducting the analysis of literature. The research was carried out in the most popular browser screen sizes (1920, 1366, 1024, 768, 480px). For each method, five tests were performed in different size web browsers fifty times each. The results were compared according to the following criteria: loading speed (s), bandwidth (KB), price and support.

SĄVOKŲ APIBRĖŽIMAS

Tinklalapio užkrovimas – tai laiko tarpas nuo užklausos išsiuntimo į serverį iki pilno visų reikiamų failų užkrovimo ir jų atvaizdavimo naršyklės lange.

Tinklalapio užkrovimo greitis – tai laikas, per kurį užfiksuojamas užkrovimas.

Užkrovimo greičio optimizavimas – tinklalapio užkrovimo laiko mažinimas taikant efektyviausius metodus.

Modernus tinklalapis – gali būti apibrėžiamas įvairiai, tačiau remiantis tyrimo tema, teigsime, jog tinklalapio dizainas – interneto puslapis, kurio HTML failas naudoja CSS stiliaus aprašymą, yra įterptas bent vienas „JavaScript“ scenarijus ir (arba) įdiegtos PHP funkcijos. Tinklalapio struktūra keičiasi priklausomai nuo naršyklės įstrižainės [1].

Domenas – interneto adresas, kuris nukreipia į nurodytą serverį.

Pralaidumas (angl. *bandwidth)* – iš serverio išsiunčiamos informacijos kiekis bitais [2].

Grafinis elementas – paveikslėlis (angl. *image*), kuris perteikiamas statiniu vaizdu. Tyrimui naudojamų failų formatai .jpeg ir .png.

DNS (ang. *Domain Names Server*) – internetinių adresų serveris, kuriame saugomi visi internetiniai adresai (domenai) ir IP adresai, į kuriuos nukreipiama norint pasiekti turinį.

Naršyklė - (angl. *browser*) yra programa, skirta atvaizduoti internetinius puslapius (tinklalapius) žiniatinklyje, vidiniuose įmonės tinkluose ar savo kompiuteryje. Interneto puslapių peržiūra, dažnai naudojantis juos siejančiomis hipertekstinėmis nuorodomis, vadinama naršymu. Be HTML pagrindu sukurtų dokumentų naršyklės paprastai gali atvaizduoti ir kitokio tipo dokumentus [3].

ĮVADAS

Šiomis dienomis, sparčiai besivystant technologijoms, turime 2,6 kartais greitesnę interneto prieigą nei 2011 metais [4]. Tačiau smarkiai išaugo ir vartotojų skaičius: 2015 metais užfiksuota daugiau nei 3 mlrd. visame pasaulyje [5]. Tai rodo, jog efektyvus turinio perdavimas, esant dideliam apkrovimui, turi didelę įtaką išteklių sunaudojimui.

XXI amžiaus antrasis dešimtmetis iš svetainių pareikalavo lankstumo, t. y. prisitaikymo prie įvairių interneto vartotojų. Nerašyta taisykle tapo tinklalapio prisitaikymas prie didelio ar prie mažo ekrano (pvz., mobiliuosiuose įrenginiuose). To pasekmė – papildomi iššūkiai spartinant svetainių užkrovimą įvairaus dydžio ekranuose.

Internete galima rasti daug sprendimų, kaip optimizuoti tinklalapio užkrovimo greitį, tačiau pastarieji metodai yra efektyvūs tik tam tikrose situacijose. Taigi, didžiausią efektyvumą galima pasiekti tinklalapį optimizuojant pagal paskirtį ir keliamas prielaidas apie vartotojo poreikius.

Problema – ilgas tinklalapių, naudojančių sudėtingus dizaino sprendimus, užkrovimo greitis;

Tyrimo objektas – grafinių elementų užkrovimas skirtingų dydžių ekranuose;

Tikslas – grafinių elementų užkrovimo metodų palyginimas įvairaus dydžio naršyklėse;

Uždaviniai:

1. Atlikti literatūros apžvalgą apie tinklalapio koduojamus elementus, grafinių elementų atvaizdavimui naudojamus metodus (interneto naršyklėje);
2. Suformuoti reikalavimus prototipui, realizuojančiam skirtingus grafinių elementų pateikimo metodus;
3. Realizuoti prototipą pagal pasirinktus reikalavimus;
4. Atlikti eksperimentus naudojant pasirinktus testavimo kriterijus.

1. LITERATŪROS ANALIZĖ

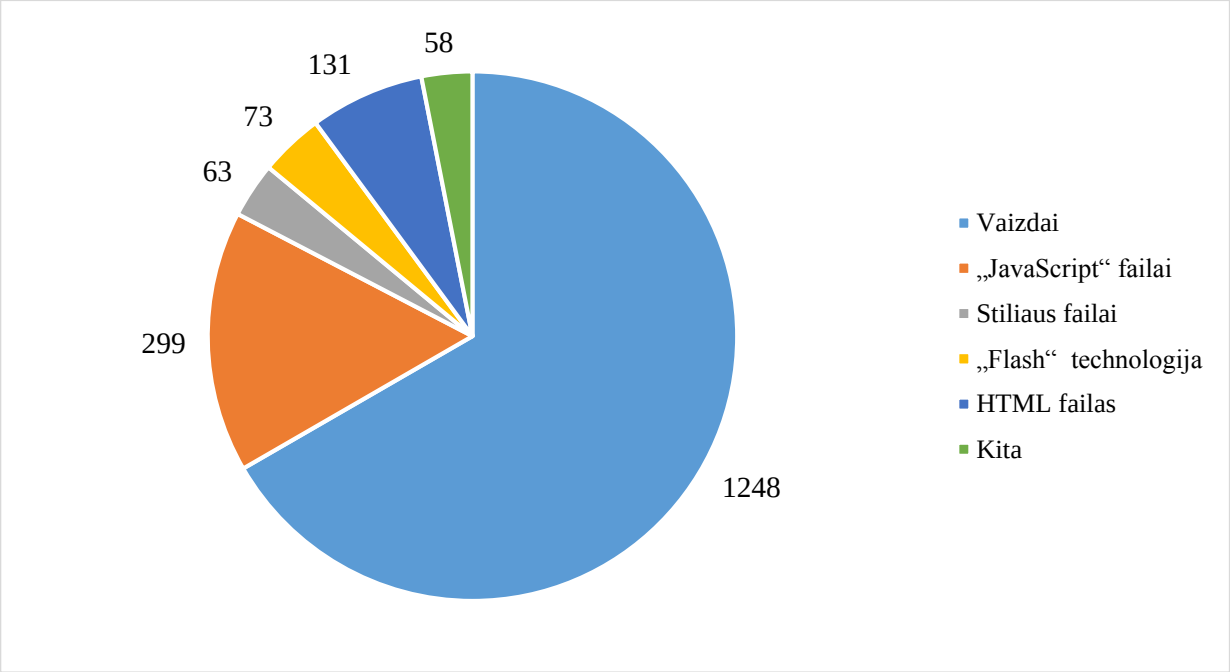
1.1. Tinklalapio dizaino elementai

1.1.1. HTML failo funkcija

HTML (angl. *HyperText Markup Language*) yra standartinė žymėjimo kalba, naudojama kuriant interneto svetaines. Interneto naršyklė gali skaityti HTML failus ir atvaizduoti juos į regimus arba girdimus tinklalapio elementus [5]. Naršyklė nerodo HTML žymių ir scenarijaus, kurie turi įtakos tinklalapio interaktyvumui, tačiau naudoja juos puslapio turiniui atvaizduoti. Tai suteikia galimybę atvaizduoti struktūrizuotus dokumentus su antraštėmis, pastraipomis, sąrašais, citatomis ir kitais elementais [3].

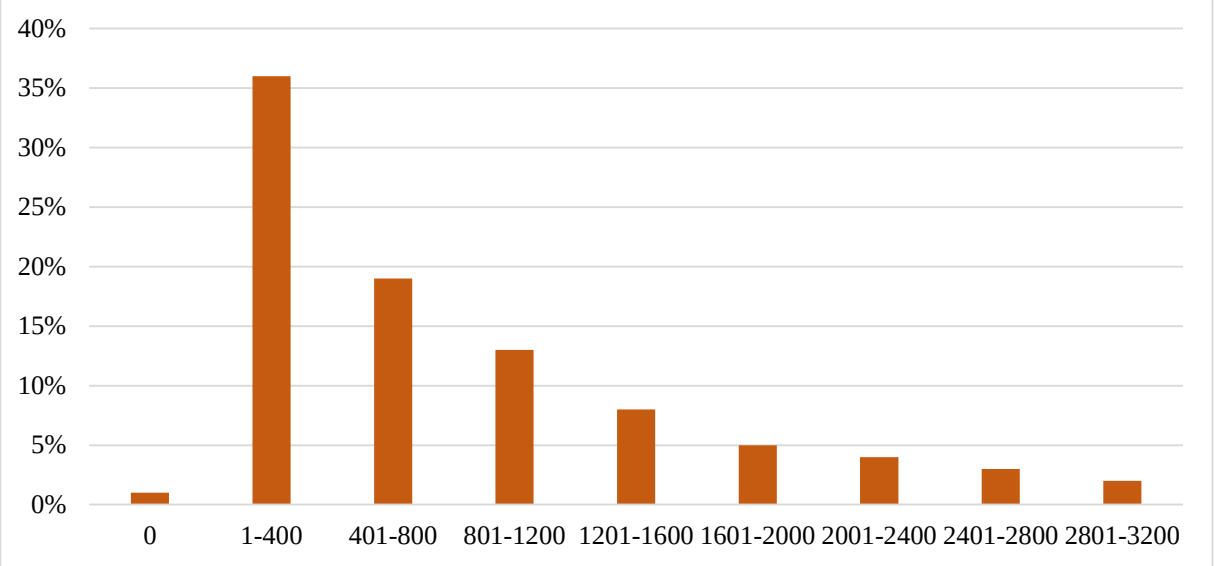
1.1.2. Grafiniai elementai

Paveikslėliai (angl. *images*) – statistiškai daugiausiai atminties užimantys failai, vertinant bendrą failų panaudojimą puslapyje (žr. 1.1. pav.). Pagal vidurkį paveikslėliai užima net 63,7 proc. visų užkraunamų puslapio failų [6].

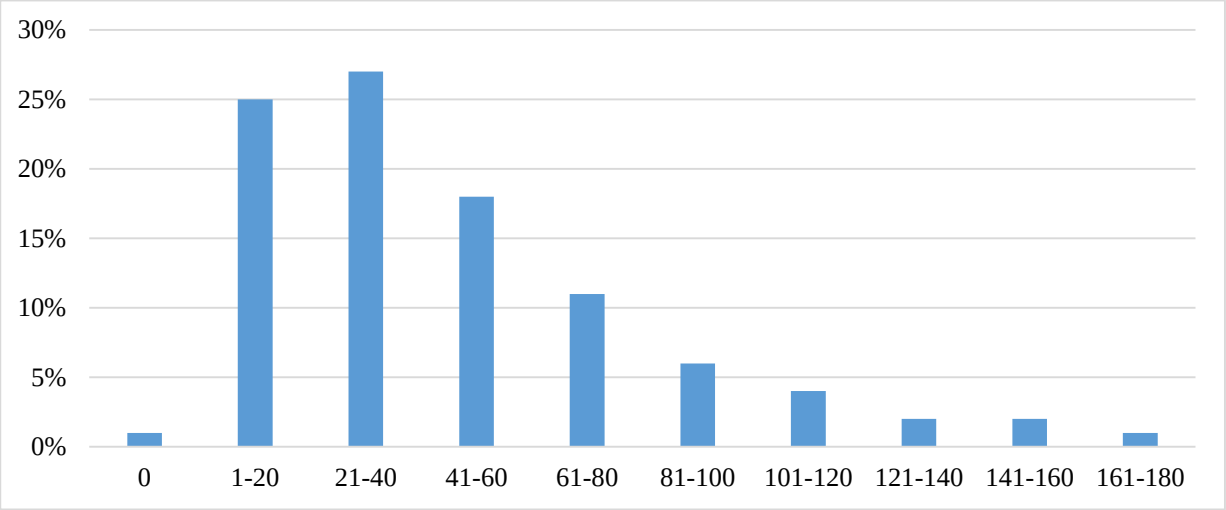


1.1. pav. Grafinių elementų dalis standartinio internetinio puslapio failuose (KB)

Tai lemia ne tik pačiame faile saugomos informacijos kiekis, bet ir pačių elementų kiekis, pateikiamas internetiniame puslapyje. Apie 36 proc. visų tinklalapių naudoja 1 – 400 KB grafinių elementų (žr. 1.2. pav.). Iš visų tinklalapių 27 proc. užkrauna 20 – 40 grafinių elementų (žr. 1.3. pav.).



1.2. pav. Grafinių elementų duomenų srauto vidurkis tinklalapyje (KB)



1.3. pav. Grafinių elementų naudojimo vidurkis viename tinklalapyje

HTML faile nurodžius grafinio elemento vietą galima atvaizduoti paveikslėlį naršyklės lange [5]. Grafiniam elementui atvaizduoti naudojamos žymos:

```
<img src="http://www.webname.com/image_location.jpg">
```

Tai yra pagrindinis grafinių elementų įterpimo metodas, tačiau yra atvejų, kada autorius nori išreikšti tą patį grafinį elementą naudodamas skirtingus pateikimo metodus dėl:

- ekrano įstrižainės;
- ekrano rezoliucijos;
- ekrano pozicijos (kompiuterio ekrano horizontali arba vertikali pozicija);
- prieigos prie interneto (gali skirtis interneto sparta, pralaidumas ar net kainodara);

Tokiu atveju tinklalapis pritaikomas įvairiems atvejams naudojant žymas, turinčias papildomas funkcijas. Jos bus aptariamos kitoje skiltyje – „Prisitaikantis dizainas“.

1.1.3. Stiliaus failai

CSS (angl. *Cascading Style Sheets*) – stiliaus failas, padedantis aprašyti dokumento išvaizdą ir formatą, parašytas žymių kalba. Failo formato pavadinimas – .css. CSS yra naudojamas kartu su HTML [7].

CSS naudojamas siekiant atskirti dokumento turinį nuo jo pateikimo stiliaus, t. y. fono, šrifto, spalvos. Šis atskyrimas pagerina turinio prieinamumą, sumažina jo pasikartojimą, suteikia daugiau lankstumo tam, kad keli HTML puslapiai pasidalintų formatavimu nurodant atitinkamą CSS failą. Pavyzdžiui, kiekviename puslapyje antraštė turi turėti žymą. Jei tai nurodoma CSS faile, tai HTML failas užima mažiau atminties. CSS turi paprastą sintaksę ir naudoja kelis anglų kalbos raktažodžius, apibūdinančius įvairias stiliaus savybes.

Stiliaus failuose nurodomas grafinių elementų atvaizdavimas. Atstumas tarp elementų gali būti matuojamas taškais (angl. *pixels*), centimetrais ar coliais (1 colis – 96 taškai, tačiau kai kurios naršyklės ar įrenginiai nepalaiko centimetrų ar colių), procentais (elemento dydis gali kisti priklausomai nuo naršyklės lango dydžio ar kitų kintamųjų) [8].

1.1.4. „JavaScript“ elementai

„JavaScript“ (JS) – dinamiška kompiuterių programavimo kalba. Dažniausiai ji naudojama kaip interneto tinklalapio HTML failo struktūros dalis, kuri leidžia kontroliuoti naršyklę, asinchroniškai bendrauti ir pakeisti dokumento turinį, kuris yra rodomas [9].

Labiausiai paplitęs „JavaScript“ panaudojimas, įtraukiant kliento elgesį į HTML puslapius, žinomas kaip „Dynamic HTML“ (DHTML). Kai kurie šio panaudojimo pavyzdžiai:

- Atnaujinus puslapio turinį ar pateikus duomenis per serverį, naudojant AJAX biblioteką (pavyzdžiui, socialinis tinklas gali leisti atnaujinti profilį neatnaujinus puslapio);
- Puslapio elementų animacija, išblukimo bei išnykimo efektai, dydžio pasikeitimai, persikėlimai.

Kadangi „JavaScript“ veikia vartotojo naršyklėje (o ne nuotoliniame serveryje), tai leidžia reaguoti į vartotojo veiksmus greičiau, programa tampa interaktyvesnė.

1.1.5. Duomenų bazės

Duomenų bazės (angl. *databases*) – organizuotas duomenų rinkimas. Duomenys dažniausiai renkami tam, kad būtų galima modeliuoti tikrovės aspektus.

Duomenų bazės valdymo sistemos (angl. *database management systems*) – DBMS – kompiuterinė programinė įranga, kuri sąveikauja su vartotoju, kitomis programomis ir pačia duomenų baze fiksuodama ir analizuodama duomenis. Bendrosios paskirties DBMS yra skirtos duomenų bazės gautoms užklausoms tikslinti, kurti, atnaujinti ir administruoti. Gerai žinomos DBMS yra „MySQL“, „PostgreSQL“, „Microsoft SQL Server“, „Oracle“, „SAP“ ir „IBM DB2“ [10].

1.2. Užkrovimo greitis ir jo optimizavimo galimybės

1.2.1. Nukreipimų į kitus puslapius sumažinimas

Peradresavimas iškviečia papildomą HTTP užklausą, dėl to delsiamas puslapio atvaizdavimas. Kiekvienas peradresavimas gali pridėti vieną ciklą (HTTP užklausa ir atsakymas) arba iškviesti keletą papildomų ciklų (DNS, pradinis apsikeitimas TCP paketais ir TLS derybos be papildomo HTTP prašymas-atsakymas ciklo) [10].

1.2.2. Failų suspaudimas

Visos šiuolaikinės naršyklės palaiko ir automatiškai išskleidžia visų HTTP užklausų .gzip formato failus (.gzip – failo formatas ir programinė įranga, naudojama failų suspaudimui ir išskleidimui). Įgalinus .gzip formato failo suspaudimą parsiunčiamų duomenų dydis sumažėja iki 90 proc., o tai sutrumpina elementų parsiuntimo laiką, tuo pačiu serveris reikalauja mažesnio duomenų pralaidumo. Taip pat pagerėja puslapio užkrovimo laikas, kraunant pirmą kartą [11].

1.2.3. Serverio atsakymo laikas

Serverio atsakymo laikas – laiko tarpas, nuo kada naršyklė kreipiasi į serverį, prašydama pateikti HTML failą iki jo išsiuntimo užkrovimui. Ilgas tokio atsakymo laikas gali būti ilgo tinklalapio užkrovimo problema. Yra daugybė priežasčių, kodėl serveris lėtai atsako: prasta programos logika, lėtas duomenų bazės veikimas, lėtas maršruto parinkimas (angl. *routing*), sistemos (angl. *framework*), bibliotekos (angl. *libraries*), procesoriaus ar atminties išteklių trūkumas. Norint sumažinti serverio atsako laiką reikia koncentruotis į visus paminėtus veiksnius. Turint duomenis yra naudojamasi įvairiomis instrukcijomis, padedančiomis išspręsti problemą [12].

1.2.4. Elementų išsaugojimas naršyklėje

Informacijos parsiuntimas iš serverio reikalauja papildomo laiko, todėl naudingiau duomenis išsaugoti. Visi serverio atsakymai turėtų nurodyti spartinimo politiką (angl. *caching policy*), siekiant nustatyti, kurią anksčiau atsiųstą informaciją galima panaudoti esamos sesijos metu. Kiekvienas šaltinis turi pateikti aiškią spartinimo politiką, kuri atsako į šiuos klausimus: ar ištekliai gali būti įrašyti (angl. *cached*), kiek ilgai jie gali būti įrašyti, kaip dažnai failai turėtų būti atnaujinami, kai spartinimo politika baigiasi. Kai serveris grąžina atsakymą, jis turi pateikti „Cache-Control“ ir „Etag“ antraštes [13]:

- „Cache-Control“ apibrėžia, kiek ilgai failas gali būti išsaugotas naršyklėje ir kitose tarpinėse saugyklose;
- „Etag“ suteikia galiojimo patvirtinimo raktą, kuris automatiškai atsiunčiamas į naršyklę siekiant patikrinti, ar informacija pasikeitė, kada paskutinį kartą ji buvo prašoma.

1.2.5. Išteklių optimizavimas (HTML, CSS ir „JavaScript“)

Išteklių optimizavimas (angl. *Resourses Minification*) – nereikalingų arba besikartojančių duomenų pašalinimas nepaveikiant turinio veikimo naršyklėje (pvz. kodo komentarai ir formatavimas, nenaudojamo kodo pašalinimas, trumpesnių kintamųjų, funkcijų pavadinimų naudojimas ir pan.) [14].

1.2.6. Paveikslėlių optimizavimas

Grafiniai elementai sudaro didžiąją dalį parsiunčiamų baitų tinklalapyje. Todėl optimizuoti vaizdai sutaupo didelę dalį serverio atminties bei pagreitina tinklalapio užkrovimą: kuo mažiau baitų naršyklė atsisiunčia, tuo mažiau duomenų srauto reikalauja klientas ir naršyklė gali greičiau parsiųsti turinį ir jį atvaizduoti ekrane.

Ieškant optimalaus formato ir optimizavimo strategijos, vaizdo turinį reikia išanalizuoti įvairiais aspektais: kokiu duomenų tipu yra užkoduotas failas, kokios yra vaizdo formato galimybės, vaizdo kokybės parametrai, rezoliucija ir pan. Be to, reikia įvertinti, ar kai kuriuos grafinius elementus verta pateikti vektoriniu būdu, jei norimą rezultatą galima pasiekti naudojant CSS sintaksę. Taip pat svarbu išsiaiškinti, kaip tinkamai pristatyti grafinius failus kiekvienam prietaiso vartotojui (mobilūs, stacionarūs įrenginiai) [15].

Dabar galima rasti daug įrankių, kurie, nesumažindami paveikslėlio kokybės, matomos žmogaus akimi, gali ženkliai sumažinti užimamos atminties dydį.

1.2.7. CSS pristatymo optimizavimas

Prieš turinio atvaizdavimą naršyklėje ji turi apdoroti visą stiliaus aprašą ir tinklalapyje esančią išdėstymo informaciją. To rezultatas – naršyklė blokuoja atvaizdavimą, kol atsiunčiami ir tvarkomi išorės stiliaus failai, kurie gali pareikalauti kelių užklausų ir taip vilkinti laiką iki pirmojo puslapio atvaizdavimo.

Jei išorės CSS failai yra maži (užimantys mažai bitų), tai stiliaus aprašymą galima įterpti tiesiogiai į HTML dokumentą (angl. *inlining*). Įterpus trumpą CSS fragmentą naršyklei leidžiama sklandžiai tęsti krovimą. Esant dideliems CSS failams reikia nustatyti ir įterpti CSS aprašymą, teikiant pirmenybę atvaizduojamam turiniui, o likusių stiliaus aprašymų užkrovimą atidedant iki pareikalavimo.

Pavyzdys, kaip įterpiami trumpi CSS failai:

Jei HTML dokumentas atrodo taip:

```
<html>
 <head>
  <link rel="stylesheet" href="small.css">
 </head>
 <body>
  <div class="blue">
   Hello, world!
  </div>
 </body>
</html>
```

O mažas CSS taip (small.css):

```
 .yellow {background-color: yellow;}
 .blue {color: blue;}
 .big { font-size: 8em; }
 .bold { font-weight: bold; }
```

Tuomet galima pritaikyti „inlining“ optimizavimo metodą:

```
<html>
 <head>
  <style>
   .blue{color:blue;}
  </style>
  </head>
 <body>
  <div class="blue">
   Hello, world!
  </div>
  <script>
   var cb = function() {
    var l = document.createElement('link'); l.rel = 'stylesheet';
    l.href = 'small.css';
```

```
    var h = document.getElementsByTagName('head')[0]; h.parentNode.insertBefore(l, h);
  };
  var raf = requestAnimationFrame || mozRequestAnimationFrame ||
    webkitRequestAnimationFrame || msRequestAnimationFrame;
  if (raf) raf(cb);
  else window.addEventListener('load', cb);
 </script>
</body>
</html>
```

Svarbiausi stiliaus aprašymo fragmentai turinio atvaizdavimui yra įterpiami viršuje, prieš informacijos atvaizdavimą. Pilnas small.css yra užkraunamas po turinio atvaizdavimo puslapyje. Jo stilius yra pritaikomos tinklalapyje, kai baigiamas užkrovimas, be turinio blokavimo [16].

1.2.8. „JavaScript“ elementų integravimas

Pridėtos „JavaScript“ elementų nuorodos gali kelis kartus prailginti užkrovimo laiką dėl išorinio naršyklės blokavimo. Jei „Javascript“ elementas yra nedidelis, rekomenduojama jį integruoti į HTML žymes [17]. Pvz.:

```
<html>
 <head>
  <script type="text/javascript" src="small.js"></script>
 </head>
 <body>
  <div>
   Hello, world!
  </div>
 </body>
</html>
```

small.js faile yra funkcija, kuri integruojama į HTML žymes:

```
<html>
 <head>
  <script type="text/javascript">
   /* small.js failo turinys */
  </script>
 </head>
 <body>
  <div>
   Hello, world!
  </div>
 </body>
</html>
```

1.2.9. Pirmenybės suteikimas matomiems elementams

Standartiškai naršyklė užkrauna visą tinklalapio langą iš karto, nebent nurodoma, kurią puslapio dalį užkrauti pirmiau. Kadangi krovimosi metu pirmiausia pastebima viršutinė tinklalapio dalis, galima inicijuoti šios dalies užkrovimą. Tam reikia sudaryti puslapio struktūrą:

- jei HTML pirmiau užkrauna trečios šalies (nereikšmingus) blokelius anksčiau nei pagrindinę informaciją, reikia sukeisti juos vietomis;
- jei svetainė sudaryta iš dviejų stulpelių, kur viename yra navigacija (kategorijų sąrašas), o kitame – straipsnio turinys, tai straipsnio užkrovimui turėtų būti nurodomas aukštesnis prioritetas.

1.2.10. Užkrovimas pagal pareikalavimą

Yra tinklalapių, kuriuose, net ir atlikus optimizavimą, užkraunamos informacijos kiekis siekia 5 – 10 MB. Dažnai didelė dalis informacijos lieka neperžiūrėta. Todėl tokiais atvejai yra naudojamas užkrovimo metodas pagal pareikalavimą. Šį greito užkrovimo metodą naudoja dauguma žiniasklaidos portalų. Tokio metodo logika panaši į prieš tai paminėtą – prioritetų suteikimas matomiems elementams. Esminis skirtumas – vartotojui turinys neužkraunamas tol, kol jo nepareikalaujama [18].

1.2.11. Turinio platinimo tinklai

Turinio platinimo tinklas (angl. *Content Distribution Network - CDN*) yra didelė paskirstyta serverių sistema visame interneto tinkle. CDN tikslas – pristatyti turinį galutiniams vartotojams didesniu prieinamumu ir aukštesne kokybe. CDN pateikia didelę dalį turinio: žiniatinklio objektus (tekstai, grafika, scenarijai), siunčiamus objektus (garso ir vaizdo failai, programinė įranga, dokumentai), programas (e-komercija, portalai), tiesiogiai transliuojamą žiniasklaidą, žiniasklaidos transliacijos pareikalavus (angl. *on-demand*) ir socialinius tinklus [11].

1.3. Tinklalapio užkrovimo analizė ir įrankiai

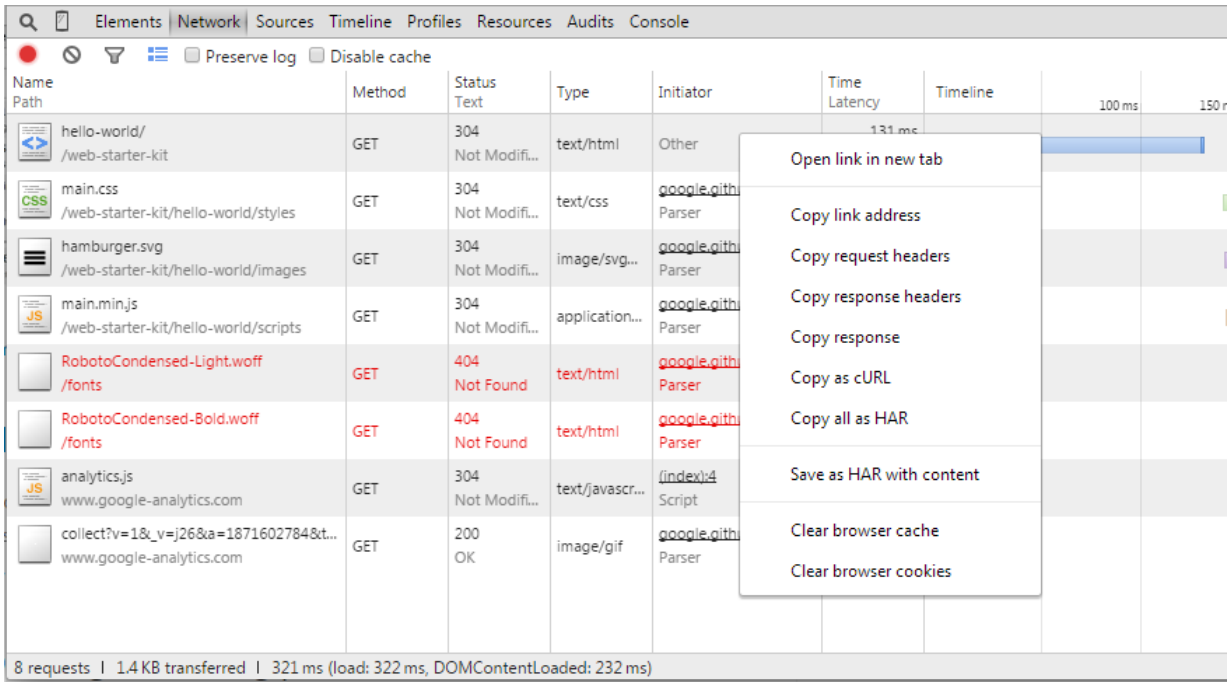
1.3.1. Apache Benchmark

AB (angl. *Apache Banchmark*) yra įrankis, kuris atlieka Apache (HTTP) serverio analizę. Šis įrankis gauna „Apache“ instaliacijos veiklos rezultatus. Analizė pateikia atsakymą, kiek užklausų per sekundę serveris yra pajėgus perduoti.

AB įrankis yra siejamas su standartiniu Apache kodo platintoju ir, taip pat kaip Apache serveris, yra nemokama atviro kodo programinė įranga, platinama Apache licencijos sąlygomis.

1.3.2. Chrome DevTools

Chrome Developer Tools (toliau DevTools) yra tinklalapių kūrimo ir derinimo įrankis, sukurtas Google Chrome rinkinyje. DevTools suteikia galimybę pasiekti tinklalapio užsikrovimo rezultatus iš savo naršyklės (žr. 1.4. pav.). DevTools įrankis skirtas efektyviai susekti blokų stiliaus aprašymus, nustatyti „JavaScript“ ribines vertes ir gauti įžvalgų kodo optimizavimui [12]



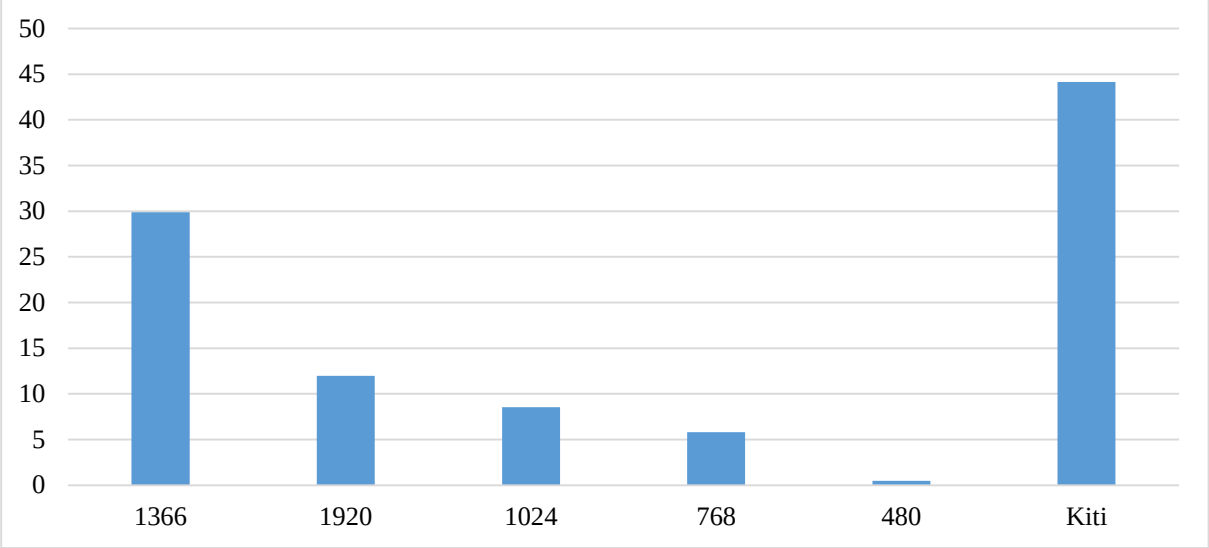
1.4. pav. Elementų užklausos ir užkrovimas realiuoju laiku

Networks – DevTools įrankio dalis, kuri suteikia informaciją apie elementus, kurie yra užklausiami ir parsiunčiami internetu realiuoju laiku. Identifikuojant užklausas, kurios yra kraunamos ilgesnį laiką nei numatyta, galima optimizuoti užkrovimą.

1.4. Prie ekrano dydžio prisitaikantis dizainas

Reaguojantis, prisitaikantis dizainas (angl. *responsive*) – tai tinklalapio pritaikymas įvairiems naršyklių ekrano dydžiams bei mobiliesiems įrenginiams. Tokiu būdu vartotojo naudojimasis turiniu tampa paprastesnis, reikalauja mažiau slankiojimo, priartinimo ir kitų papildomų veiksmų [13].

Natūralu, jog pritaikant tinklalapio vaizdą naršyklei, orientuojamasi į dažniausiai naudojamus ekranus. Remiantis pasauliniais rodikliais (žr. 1.5. pav.) dažniausiai tinklalapiai atidaromi 1366px pločio ekranuose [14].



1.5. pav. Ekranų rezoliucijos taškais pagal naudojimą (proc.)

Dažniausiai didelės raiškos ekranų plotis sudaro 1920px, o mažos rezoliucijos ekranai turi 480 ir 360 taškų skiriamąją gebą vertikalioje ašyje. Remiantis šiais duomenimis tyrime bus naudojami vaizdų pritaikymai šiems naršyklių ekranams.

1.4.1. Prisitaikantys paveikslėlių dydžiai

Stacionariųjų kompiuterių bei mobiliųjų įrenginių ekranų dydžiai skiriasi, todėl mažėjant ekrano įstrižainei paveikslėlis turėtų būti mažinamas [15]. Ekrano įstrižainės duomenis nuskaito .html faile patalpinta žyma:

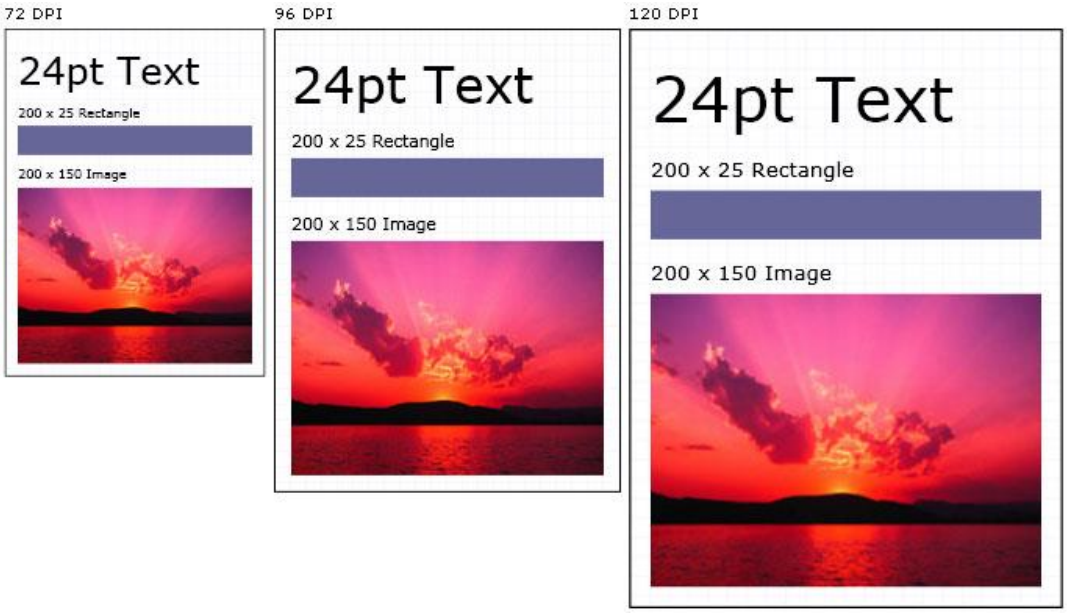
```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Paveikslėlis pritaikomas naršyklės pločiui naudojant įrašą CSS faile:

```
img {max-width: 100%;}
```

Paveikslėlis bus pateikiamas 100 proc. naršyklės plotyje, todėl reikia įsitikinti, kad paveikslėlio kokybė yra gera, kad būtų pateikiama didelės įstrižainės ekranuose.

Tobulėjant mobiliųjų ir stacionariųjų įrenginių ekranams didėja jų raiška, kas reiškia, jog į vieną kvadratinį centimetrą sutalpinama vis daugiau taškų. Tai ypač būdinga mobiliųjų įrenginių ekranams, todėl yra sukurtas papildomas matas – įrenginio taškų santykis (angl. *device-pixel-ratio*) (žr. 1.6. pav.). Device pixel ratio yra santykis tarp fizinių taškų skaičiaus ir taškų intensyvumo (angl. *device-independent pixels*) [16].



1.6. pav. Įrenginio ekrano taškų santykis

Esant šiam parametrui galima nurodyti skirtingus paveikslėlių šaltinius esant skirtingam įrenginio taškų santykiui, kuris nurodomas kaip *x* deskriptorius:

```
<img src="/uploads/100-marie-lloyd.jpg"
srcset="/uploads/150-marie-lloyd.jpg 1.5x, /uploads/200-marie-lloyd.jpg 2x"
alt="" width="100" height="150">
```

Žymė *„src“* nurodo standartinio paveikslėlio šaltinį, kuris atvaizduojamas tipiniu taškų santykiu įrenginių ekranuose arba naršyklėse, kurios nepalaiko *„srcset“* žymės.

```
<img sizes="100vw" srcset="wolf-400.jpg 400w, wolf-800.jpg 800w, wolf-1600.jpg 1600w" src="wolf-400.jpg"
alt="The rad wolf">
```

Šio pavyzdžio paveikslėlis užima visą ekrano plotį, kurį nuskaito *„viewport“* žymė. Jei vartotojo ekrano plotis yra 320 CSS taškų, tai yra ekvivalenčiai apibrėžiama:

```
wolf-400.jpg 1.25x, wolf-800.jpg 2.5x, wolf-1600.jpg 5x.
```

Iš kitos pusės, jei ekrano plotis yra 1200 CSS taškų, tai apibrėžiama:

```
wolf-400.jpg 0.33x, wolf-800.jpg 0.67x, wolf-1600.jpg 1.33x.
```

Pagal naudojamą *„w“* deskriptorių ir *„size“* atributą, vartotojo agentas (angl. *user-agent*) gali pasirinkti parsisiuntimui tinkamą paveikslėlio šaltinį, nepriklausomai nuo įrenginio ekrano dydžio [17].

Nepaisant to, kad mobilieji telefonai turi stacionaraus kompiuterio ekrano rezoliuciją (kartais net didesnę), vis vien išlieka ekrano dydžių problema. Žmogaus akis sunkiau mato tos pačios

rezoliucijos objektus mobiliojo telefono nei kompiuterio ekrane. Sprendimas - <picture> žyme skirtingais atvejais pateikti ne tik skirtingos rezoliucijos paveikslėlius, bet ir apkarpytus taip, kad aiškiai matytųsi akcentuojami vaizdai. Šis būdas vadinamas meniniu nukreipimu (angl. *art direction*).

```
<picture>
        <source media="(min-width: 45em)" srcset="large.jpg">
        <source media="(min-width: 32em)" srcset="med.jpg">
        <img src="small.jpg" alt="The president giving an award.">
</picture>
```

Žymą <picture> palaiko daugelis modernių interneto naršyklių (žr. 1.1. lentelė).

1.1. lentelė. Metodo palaikymas skirtingose naršyklėse

Versijos	**Chrome**	**Firefox**	**Internet Explorer**	**Opera**	**Safari**
Stacionarus kompiuteris	38	38	Edge 13	25	9.1
Mobilus įrenginys	38	38	nepalaiko	25	9.3

Šis aprašymas atlieka tą pačią funkciją kaip ir medijos aprašymas .css faile:

```
img { width: 300px; height: 300px }
@media (min-width: 32em) { img { width: 500px; height:300px } }
@media (min-width: 45em) { img { width: 700px; height:400px } }
```

Paveikslėliai gali būti užkraunami iš trečiosios šalies. Tokiu atveju vidiniame HTML faile yra papildomų taisyklių interneto ryšiui (WiFi, 3G) [18].

1.4.2. Adaptyvūs paveikslėlių dydžiai

Adaptyvus paveikslėlių dydžių metodas nustato lankytojo ekrano dydį ir automatiškai išsaugo ir pristato reikiamo dydžio paveikslėlį bei įkomponuoja internetiniame puslapyje naudodamas PHP funkciją. Svetainei, kuri nepritaikyta mobiliesiems įrenginiams, šis metodas nereikalauja pakeisti paveikslėlio žymių HTML faile. Adaptyvaus paveikslėlio (angl. *adaptive-image*) sprendimą pasiūlė svetainių kūrėjas Matt Wilcox [19].

Metodo integravimas:

- pridėti .htaccess ir adaptive-images.php failus į pagrindinį serverio aplanką;
- įterpti „JavaScript“ į <head> vietą žiniatinklyje;
- nurodyti naršyklės lango ypatybes adaptive-image.php faile “resoliutions” skiltyje.

Metodo veikimas:

- krovimosi pradžioje HTML failas, naudodamas įrašytą „JavaScript“, nuskaito informaciją apie naršyklės ekrano dydį taškais. Ekrano parametrai yra saugomi naršyklės slapukuose;
- kada naršyklė aptinka <img> žymę, siunčia užklausą serveriui;
- serveris priima užklausą apie paveikslėlį ir tuoj pat tikrina .htaccess failą, ar yra specialių nurodymų pateikiamiems failams;
- .htaccess failas nurodo, kad esant bet kuriai .jpg, .gif ar .png formato failo užklausai, vietoj jo siųsti adaptive-image.php failą;
- adaptive-image.php failas atlieka tokias funkcijas:
    - pagal nurodytą ekrano rezoliuciją pateikia reikiamą paveikslėlį iš nurodyto aplanko;
    - jei paveikslėlio nėra, jis yra sukuriamas, išsaugomas nurodytame aplanke ir kitą kartą užkraunamas iš jo.

1.4.3. Paveikslėlių pritaikymas naudojant naršyklės slapukus

Naršyklei kreipiantis į serverį dėl failo kartu siunčiami slapukų duomenys. Šiuos duomenis galima gauti patalpinus „JavaScript“ eilutę į <head> skiltį HTML faile:

```
document.cookie = "device_dimensions=" + screen.width + "x" + screen.height;
```

Pagal gautus duomenis funkcija parenka tinkamą paveikslėlį ir siunčia iš serverio į naršyklę. Taikant šį metodą paveikslėlis HTML faile yra aprašomas tokia žyme:

```
<img src="images/?test.png">
```

Metodo veikimas:

- kraunant HTML failą „JavaScript“ pirmiausia nustato slapukus ir reikiamą <img> elementą;
- paveikslėlio žyma, naudodama užklausos eilutę, prašo reikiamo paveikslėlio;
- pagal pateiktus duomenis serveris siunčia turimą failą arba optimizuoja, išsaugo ir tik tuomet siunčia.

Jei naršyklėje slapukai ar „JavaScript“ funkcijos yra uždraustos, tada yra siunčiamas aukštos rezoliucijos paveikslėlis tam, kad dideliame ekrane vaizdas būtų kokybiškas [20].

1.4.4. Trečiųjų šalių serveriai

Paveikslėlius galima užkrauti pasitelkus trečiųjų šalių sprendimus. Grafinių elementų pateikimas iš kitų serverių nereikalauja konfigūracijų savo serveryje. Tokios paslaugos vadinamos „Prisitaikančio vaizdo apdorojimo paslaugos“ (angl. *The Responsive Image Processing Service, CDN*). Jų veikimo pagrindas:

- nurodomas vaizdo šaltinis:

```
<img data-src="http://www.your-site.co/image.jpg" class="cdn"/>
```

- trečioji šalis jį parsisiunčia ir pritaiko kiekvieno ekrano dydžiui ir rezoliucijai:

```
<script src="//treciosios-salies-JS-biblioteka/"></script>
<script>
        Javascript naudojamas paveikslėlio konfigūracijai bei nurodomas serveris iš kurio atsiunčiamas pritaikytas vaizdas.
</script>
```

- vaizdas užkraunamas pagal pareikalavimą.

Pagal standartą, serveris pateiks paveikslėlį pagal įrenginio ekrano dydį, kuris nustatomas pagal „*user-agent*“ eilutę. Mobiliesiems įrenginiams gali būti siunčiamas 480px pločio vaizdas, planšetiniams kompiuteriams 768px pločio vaizdas [21].

1.4.5. Naršyklių palaikomos funkcijos

Be funkcijų palaikymo, kurios reikalingos tam tikrų metodų veikimui, naršyklės turi žinoti duomenis apie ekraną, kuriame turi būti atvaizduojama informacija. Viena iš pagrindinių duomenų surinkimo grupių – „*media queries*“.

1.2. lentelė. Funkcijos „*media queries*“ palaikymas skirtingose naršyklėse

IE	Edge	Chrome	Firefox	Opera	Safari	iOS Safari	Opera Mini	Chrome Android	Firefox Android
Nepalaiko	13	26	3.5	6.1	10.1	7.1	8	49	45

Palaikydama CSS3 funkcijas naršyklė gali nurodyti „*media*“ grupes: all, screen, print, speech. Ši „*media queries*“ funkcija palaikoma tik naujesnėse naršyklės versijose (žr. 1.2. lentelė).

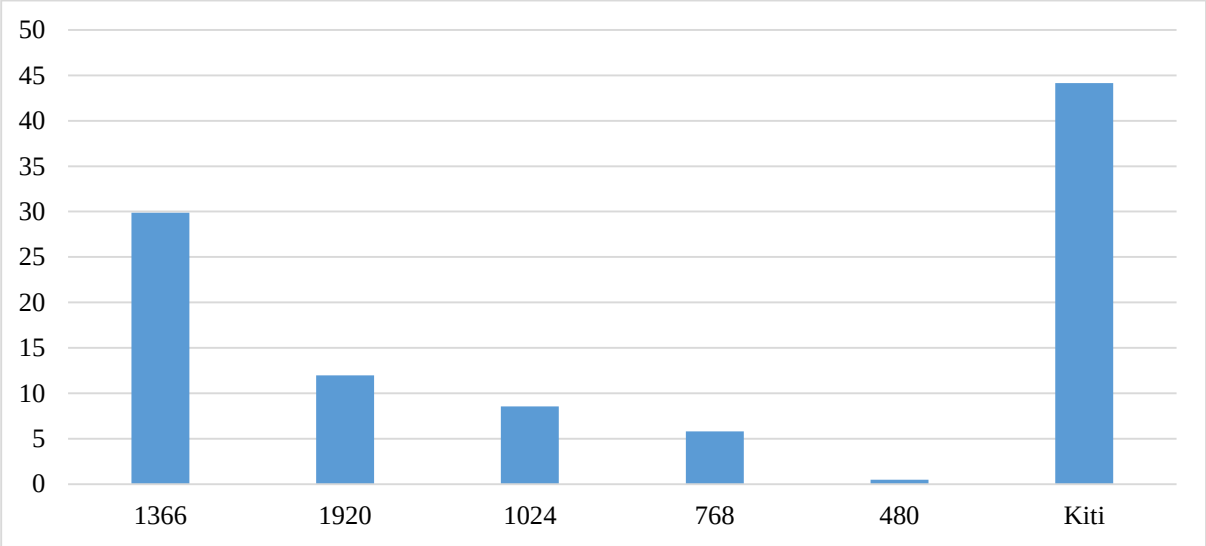
Dažniausiai naudojamos užklausos naršyklės ekranui [22]:

- width – naršyklės lango plotis;
- height – naršyklės lango aukštis;

- aspect-ratio – naršyklės lango pločio ir aukščio santykis;
- orientation – įrenginio padėtis (gulsčias ar vertikalus);
- resolution – įrenginio ekrano rezoliucija.

1.5. Ekranų dydžiai

Naudojantis internetu, naršyklės užkrauna turinį įvairaus dydžio ekranuose. Dažniausiai naudojami penki standartinio dydžio ekranai, kurie sudaro apie 56 proc. visų naudojamų ekranų dydžių (žr. 1.7. pav.). Į šią grupę patenka 1920, 1366, 1024, 768 ir 480 taškų pločio ekranai [23].



1.7. pav. Ekranų plotis procentais pagal interneto naudojimą

Remiantis šiais duomenimis, tyrimui naudojami penki dažniausiai sutinkami naršyklės ekranų pločiai. Tokiu atveju tyrimas geriausiai atitiks 2016 metų rinkos poreikius.

2. PROJEKTAVIMAS

1.6. Tiriami metodai

Tyrimo metu pasirinkti metodai yra analizuojami ir lyginami pagal lentelėje 2.1 paminėtus požymius. Pirmi trys požymiai aptarti ankstesniuose skyriuose. Metodo palaikymas vertinamas pagal tai, kokie funkciniai reikalavimai keliami populiariausioms naršyklėms (Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera [24]).

Naudojamų metodų trumpiniai

HTML5 – paveikslėliai užkraunami naudojant HTML5 funkciją. Žymose aprašoma, koks vaizdas turi būti užkraunamas tam tikrame ekrano plotyje.

PHP – paveikslėliai parenkami pagal slapukų duomenis ir sumažinami naudojant PHP funkciją, jei tokio dydžio paveikslėlio nėra serveryje.

COOKIES – paveikslėliai parenkami naudojant naršyklės slapukus. Visi grafiniai elementai laikomi viename aplanke.

RESRC – trečiosios šalies paslauga paveikslėlių pateikimui (resrc.it sprendimas).

Atliekant šių metodų tyrimą, duomenų rinkimas bus orientuotas pagal pateiktų požymių palyginimą:

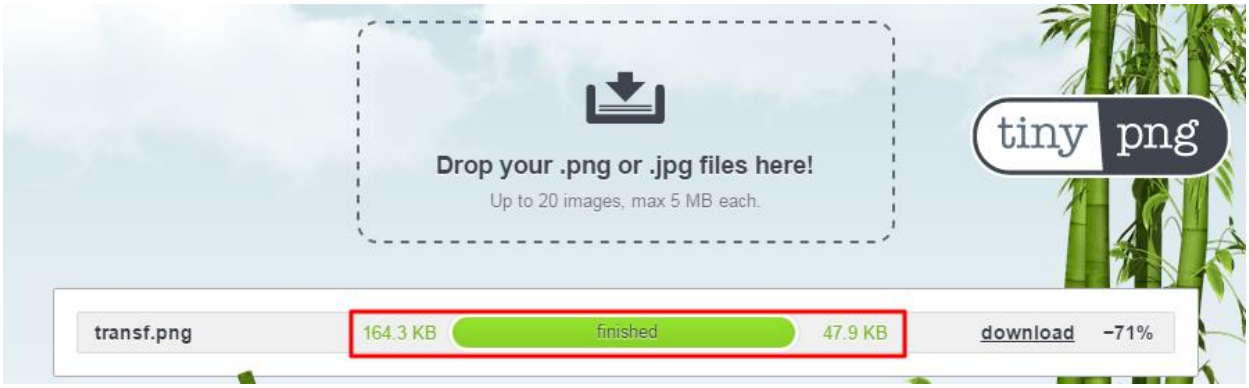
- užsikrovimo greitis – matuojamas laikas (s) nuo užklausos išsiuntimo ir pilno duomenų parsiuntimo į naršyklę;
- parsiunčiamų duomenų srautas – kiek duomenų (KB) parsiunčiama tinklalapio užsikrovimo metu;
- metodo eksploatavimo kaina – kiek kainuoja integruoti metodą ir jį eksploatuoti (palaikymo kaštai);
- skirtingų naršyklių, esančių rinkoje, palaikymas.

Požymiai pateikti prioritetų tvarka.

1.7. Paveikslėlių paruošimas

Tinklalapiuose dažniausiai naudojami grafinių elementų formatai .jpg (73,5 proc. visų grafinių elementų, naudojamų tinklalapiuose) ir .png (71,4 proc.) [25]. Panaudoti du .jpg formato failai ir vienas .png. Iš žiniatinklio parsisiųsti paveikslėliai, kurių plotis – 1920px. Iš šių paveikslėlių sugeneruoti kitų dydžių failai: 1366px, 1024px, 768px, 480px.

Visų dydžių paveikslėliai optimizuoti naudojant įrankį, kuris sumažina akimis neatskiriamų spalvų kiekį, taip sutaupant atminties tiek .jpg, tiek .png formatams (žr. 2.1 pav.) [26]. PHP ir RESRC metodams pakanka pateikti tik vieną geriausios raiškos paveikslėlį ir jie automatiškai paruošia ir pristato reikiamo dydžio grafinį elementą.



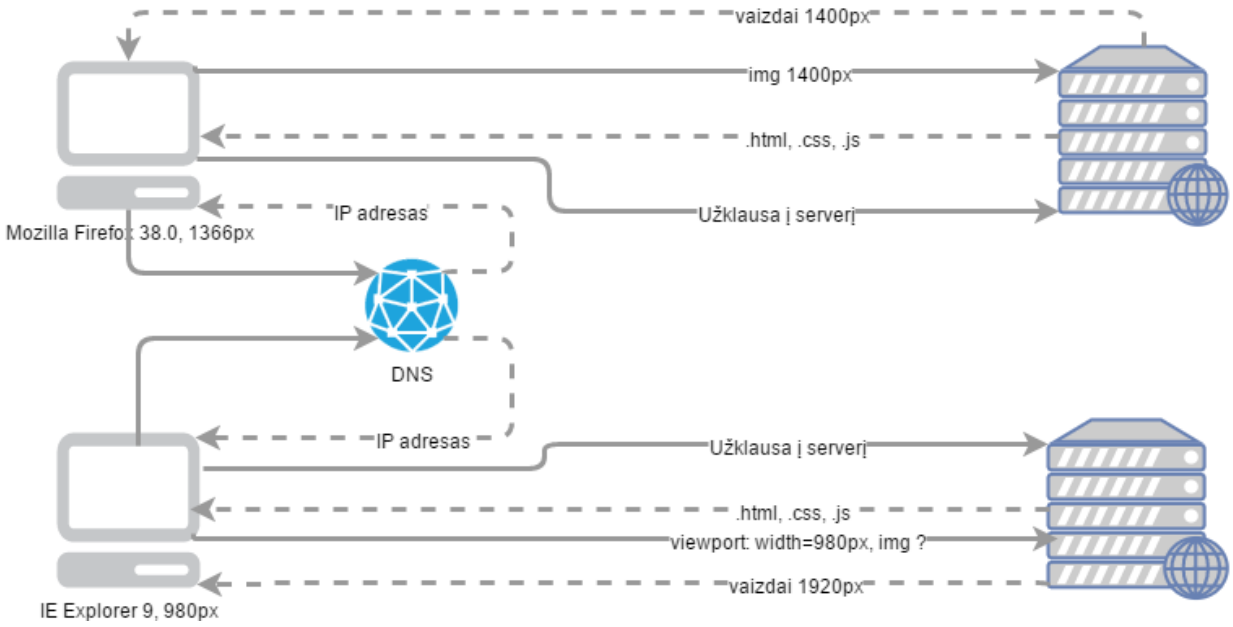
2.1. pav. Paveikslėlių optimizavimas naudojant tynyPNG sprendimą

1.8. Metodų realizavimas

Tiriami objektai (metodai) pasiekiami tuo pačiu domenu – mag.aut.lt. Metodams sukurti identiški .html puslapiai, atvaizduojantys du .jpg ir vieną .png formatų paveikslėlius. Sukurti puslapiai turi meniu, kuris naudoja jquery biblioteką ir prisitaiko prie ekrano pločio.

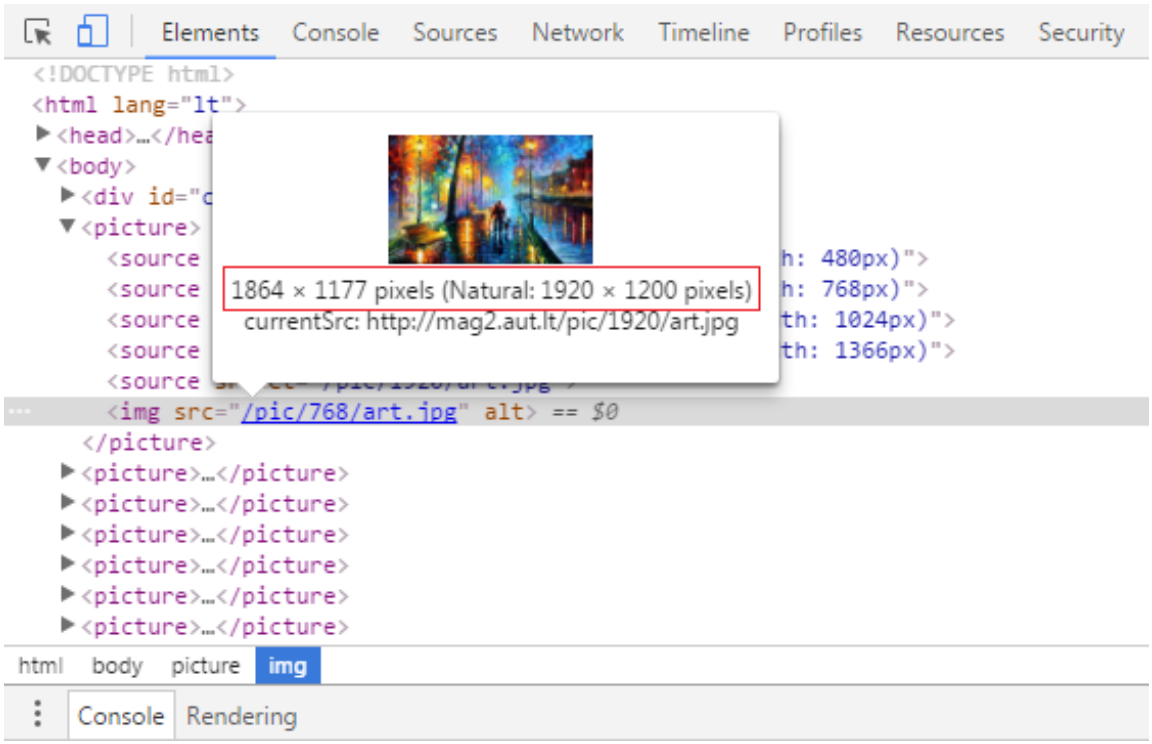
1.8.1. HTML5 metodas

Pirmas metodas naudoja html5 konsorciumo patvirtintą funkciją, kuri <picture> žymoje aprašo, kokiu atveju turi būti atsiunčiamas reikiamo dydžio paveikslėlis [27]. DNS nurodomas serveris, iš kurio turi būti atsiunčiamas .html failas ir jame nurodyti paveikslėliai. Žyma <picture> aprašo sąlygas ir nuorodas failų, kurie turi būti atsiunčiami. Parametrai aprašomi tokia pat sintakse kaip CSS faile, nurodant tuos pačius ekrano parametrus. Naršyklei nepalaikant šios funkcijos, naudojama standartinė <img> žyma ir siunčiamas šaltinyje nurodytas vaizdas.

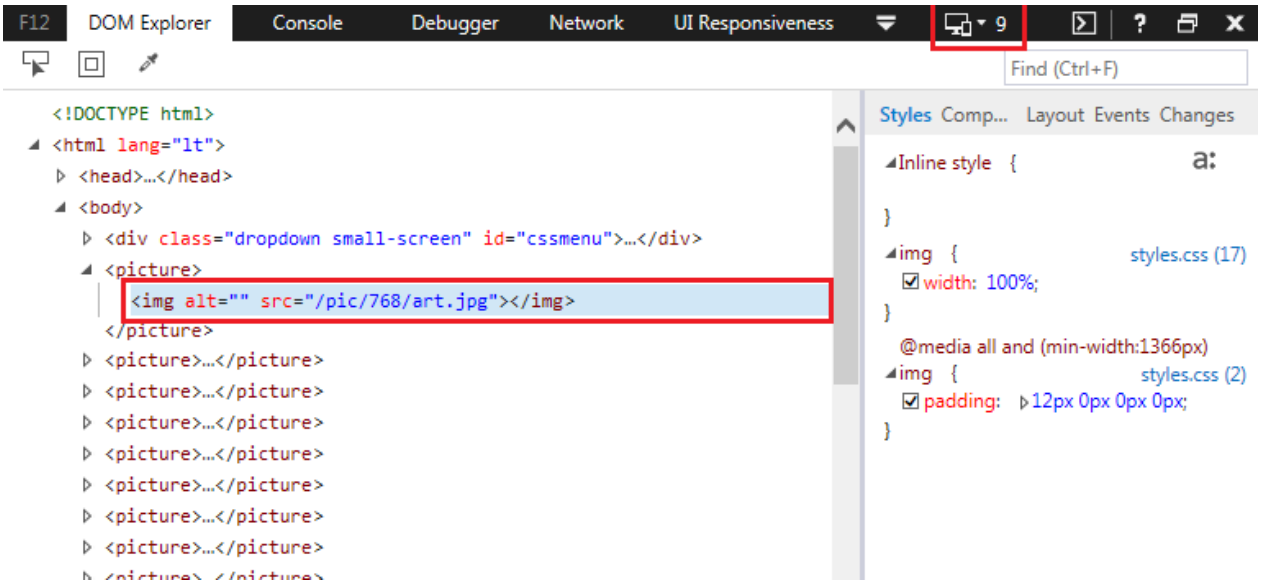


2.2. pav. Duomenų srautai vykdant paveikslėlių atsiuntimą

Schemoje (žr. 2.2 pav.) pateikti du šio metodo veikimo pavyzdžiai. Pirmuoju atveju vartotojas naudoja Mozilla Firefox 38.0 versijos naršyklę ir tinklalapį bando pasiekti per 1400px pločio langą. Suvedus internetinį adresą vykdoma standartinė procedūra. Naršyklė pagal suvestą domeną kreipiasi į DNS, o šis nukreipia į serverį, kuriame patalpintas tinklalapio turinys. Serveris, gavęs užklausą, išsiunčia .html failą vartotojo naršyklei ir atsiųstas failas yra atvaizduojamas. Pradėjus užkrovimą ir pateikus pareikalavimą atsiunčiami likę failai. HTML dokumente yra įterpta žyma, surenkanti informaciją apie naršyklės langą, todėl pasiekus <picture> žymą yra ieškoma sąlygos, atitinkančios surinktus naršyklės parametrus pagal <source> žymą. Kadangi pirmuoju variantu naršyklės plotis yra iki 1920px, todėl yra atsiunčiamas 1920px pločio paveikslėlis (žr. 2.3 pav.). Antruoju atveju Internet Explorer 9 versija nepalaiko HTML5 funkcijos (<picture> žymos), todėl jai atsiunčiamas standartinis 768px pločio paveikslėlis (žr. 2.4 pav.).

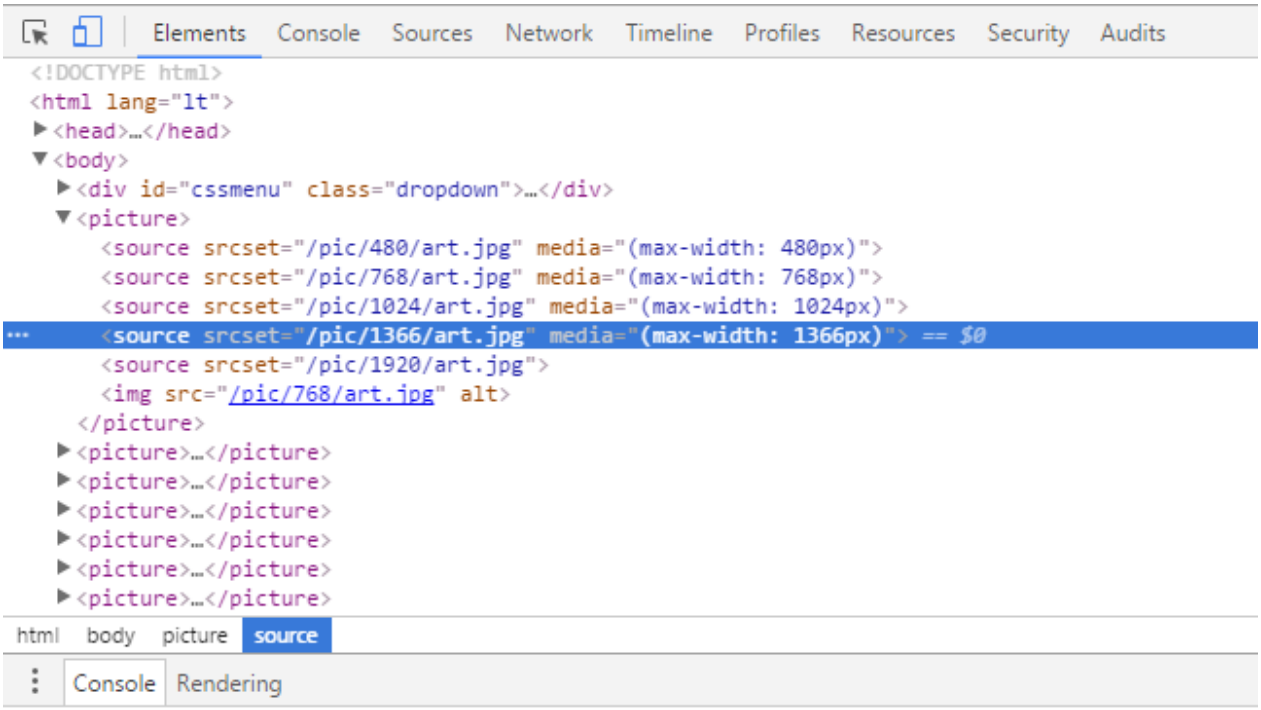


2.3. pav. Atvaizduojamas paveikslėlis pagal naršyklės lango dydį



2.4. pav. HTML5 funkcija IE naršyklėje (versija 9.0)

Naršyklei įvykdžius funkciją vietoj nurodyto šaltinio „scr“ sugeneruojama „currentSrc“ nuoroda į paveikslėlį, atitinkantį nurodytus parametrus. Jei parametrai pasikeičia neperkrovus puslapio, tai paveikslėliai užkraunami iš naujo, pagal nustatytus „media queries“ duomenis (žr. 2.5. pav.).



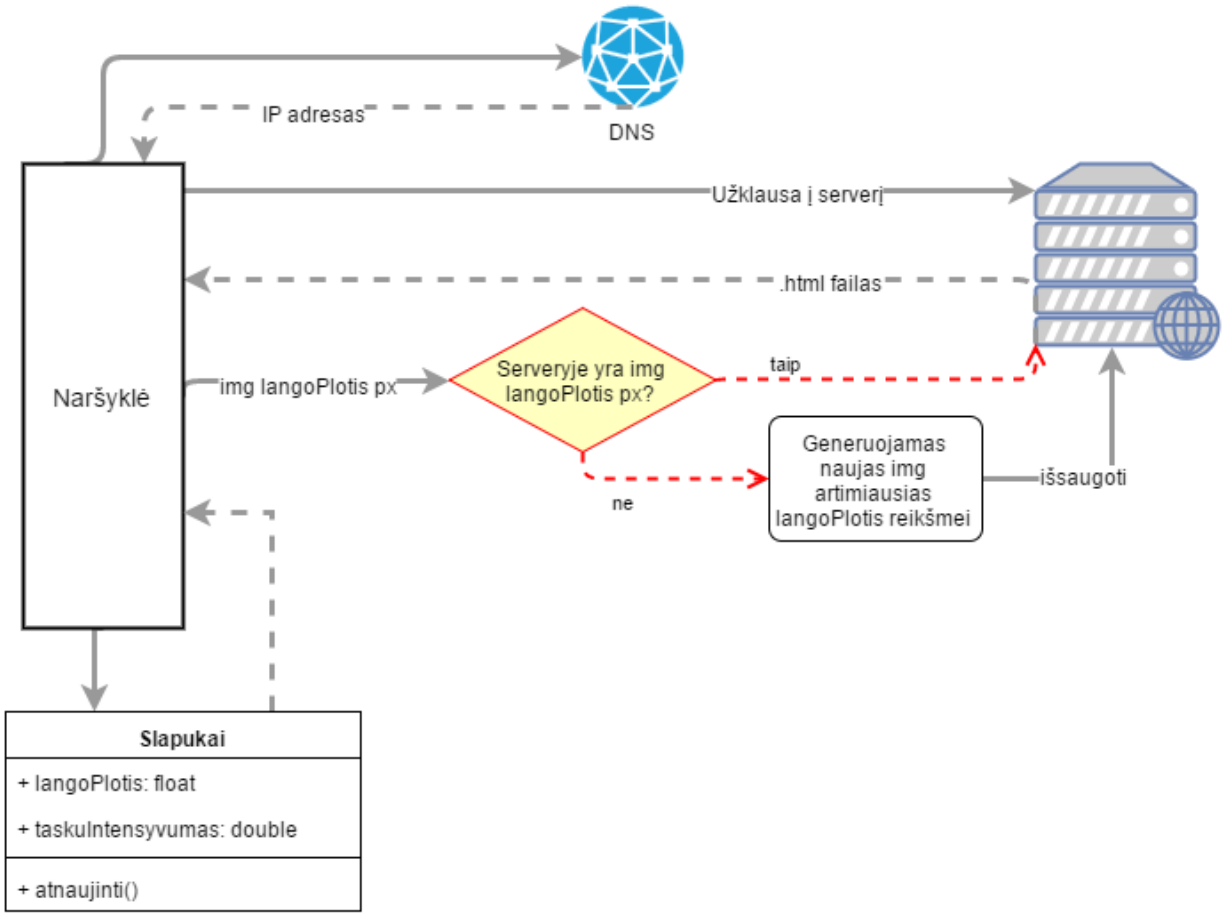
2.5. pav. Atvaizduojamas paveikslėlis, esant naršyklės langui iki 1366px

1.8.2. PHP metodas

Šiam metodui realizuoti sukurtas identiškas .html puslapis, kuriame sudėtos tų pačių vaizdų <img> žymos ir nurodyti didžiausios rezoliucijos vaizdų šaltiniai (1920px). Sektoriuje <head> įterpta „Javascript“ funkcija, kuri kuria slapukus naršyklės ekrano pločio ir taškų intensyvumo informacijai kaupti.

```
$resolutions   = array(1920, 1366, 1024, 768, 480); // ribos, ties kuriomis keičiamas paveikslo dydis (ekrano rezoliucija, taškais)
$cache_path    = "ai-cache"; // aplankas, kur talpinami sugeneruoti paveikslėliai.
$jpg_quality   = 80; // kokybė generuojamiems .jpg formato failams, nuo 0 iki 100.
```

Bandymuose, naudojant šį metodą, pasirinkta sugeneruotus vaizdus išsaugoti 80 proc. kokybe.

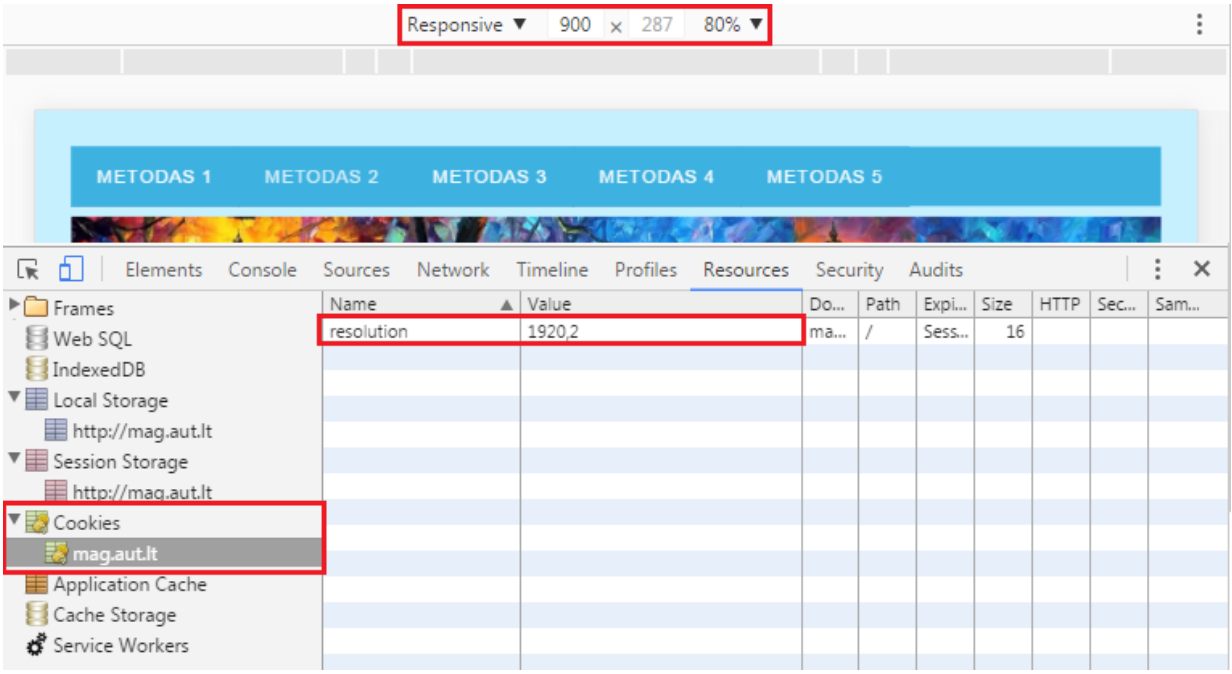


2.6. pav. PHP metodo veikimas

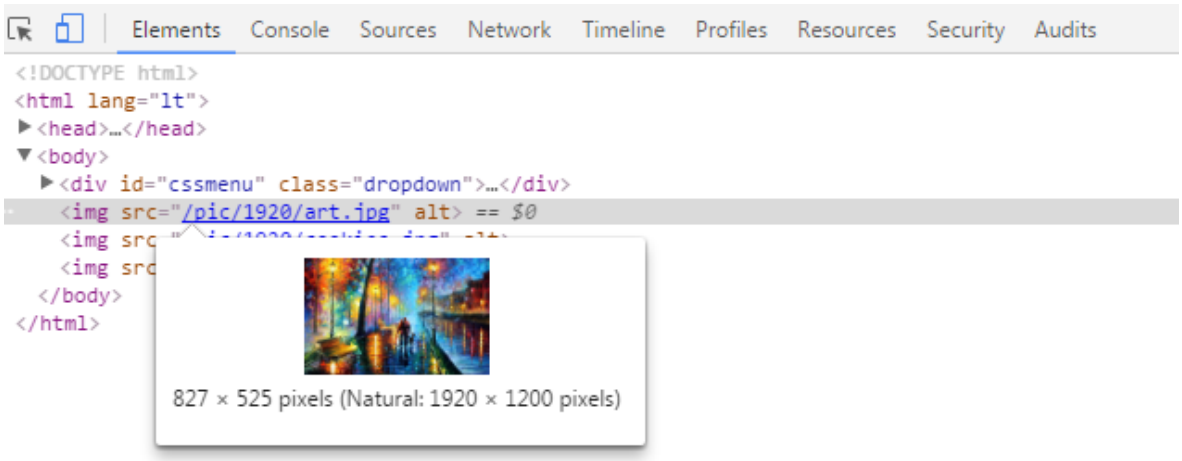
Į serverį siunčiama naršyklės lango informacija ir naudojamas PHP modulis, kuris patikrina ir parenka reikiamą dydį arba sukuria naujo dydžio paveikslėlį naujame aplanke. Jei kitą kartą kraunant puslapį pareikalaujama to pačio dydžio paveikslėlio, tai jis paimamas iš jau sugeneruotų grafinių elementų aplanko (žr. 2.6. pav.).

2.7. pav. „Javascript“ funkcijos įterpimas informacijos gavimui apie naršyklės ekrano parametrus

Naršyklės lango parametrai saugomi naršyklės istorijoje (žr. 2.7. pav.), todėl sumažinus naršyklės langą ir jį atnaujinus, kito dydžio grafinis elementas neužkraunamas, kol neatsinaujina slapukų saugoma informacija. Pašalinus senus duomenis (angl. *clear cache*) užkraunamas ankstesniam ekranui pritaikytas vaizdas (žr. 2.8 pav. ir 2.9. pav.).



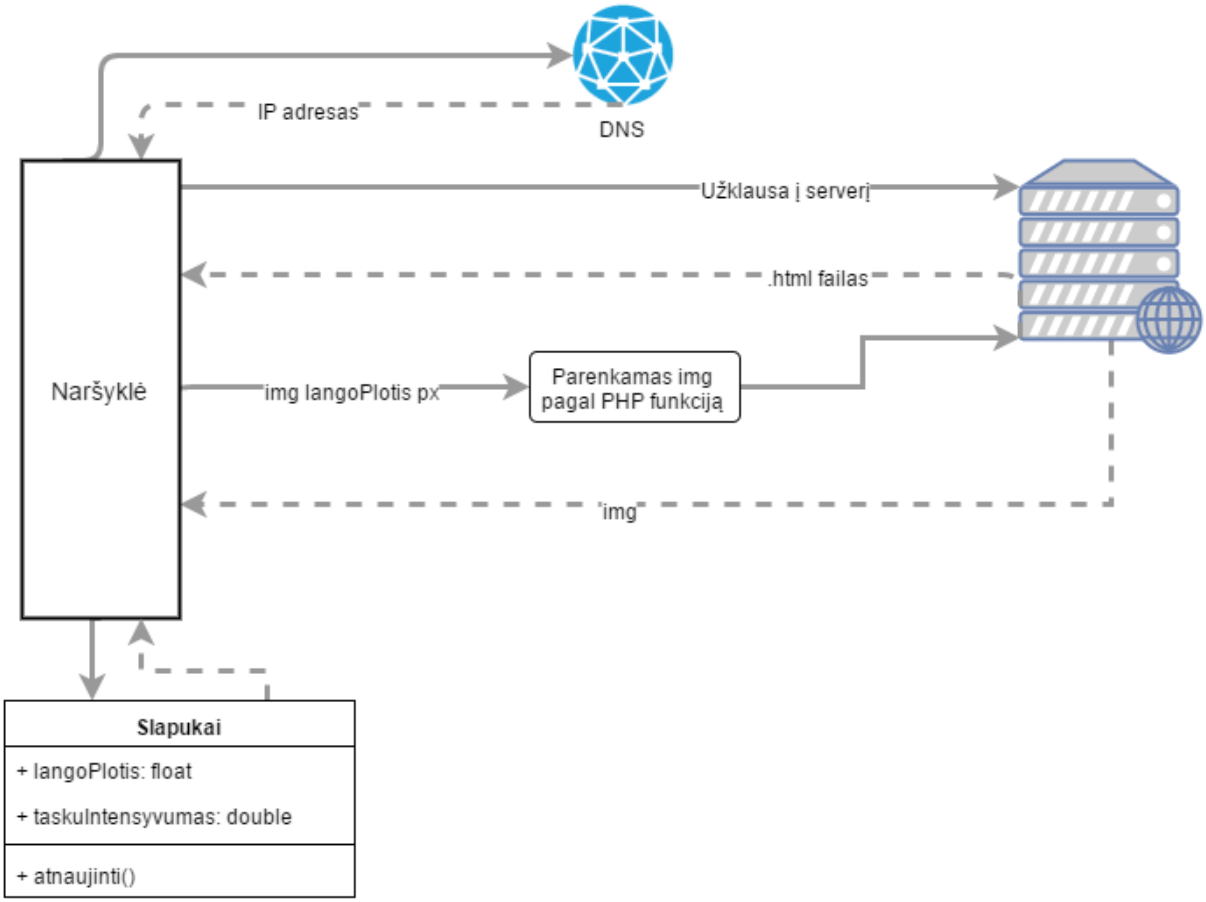
2.8. pav. Naršyklės lango parametrų saugojimas



2.9. pav. Pasikeitus naršyklės langui paveikslėlis užkraunamas pagal naršyklės slapukų informaciją

1.8.3. COOKIES metodas

Šiam metodui realizuoti taip pat įterpta „Javascript“ funkcija, kuri slapukams pateikia informaciją apie naršyklės langą ir ekrano taškų intensyvumą (žr. 2.10. pav.).



2.10. pav. Paveikslėlio parinkimo schema pagal naršyklės lango plotį

Šis metodas, skirtingai nei PHP, negeneruoja skirtingų dydžių paveikslėlių. „Javascript“ funkcijai aptikus naršyklės parametrus (žr. 2.11. pav.) yra siunčiama užklausa į serverį. Jame, pagal index.php faile (žr. 2.12. pav.) parinktus nustatymus, parenkamas reikiamo dydžio paveikslėlis ir jis atvaizduojamas naršyklėje.

```
<!DOCTYPE html>
<html lang="lt">
▼<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="styles.css">
    <script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>
    <style type="text/css"></style>
    <script src="script.js"></script>
    <script>document.cookie = "device_dimensions=" + screen.width + "x" + screen.height;</script>
    <title>Magistrinis IMF</title>
  </head>
▼<body>
  ▶<div id="cssmenu" class="dropdown">…</div>
```

2.11. pav. Slapukams pateikiama informacija

Filename	Filesize	Filetype	Last modified	Permi
art-1024.jpg	128.876	JPEG image	2016.01.10 11:5...	0644
art-1366.jpg	241.097	JPEG image	2016.01.10 11:5...	0644
art-480.jpg	83.984	JPEG image	2016.01.10 11:5...	0644
art-768.jpg	107.787	JPEG image	2016.01.10 11:5...	0644
art.jpg	384.415	JPEG image	2016.01.10 11:5...	0644
cookies-1024.jpg	35.536	JPEG image	2016.01.09 16:3...	0644
cookies-1366.jpg	77.527	JPEG image	2016.01.09 16:3...	0644
cookies-480.jpg	42.597	JPEG image	2016.01.09 16:3...	0644
cookies-768.jpg	28.914	JPEG image	2016.01.09 16:3...	0644
cookies.jpg	72.332	JPEG image	2015.12.17 15:2...	0644
index.php	1.405	PHP File	2016.01.10 11:5...	0644
transf-1024.png	192.028	PNG image	2016.01.09 16:3...	0644
transf-1366.png	329.032	PNG image	2016.01.09 16:3...	0644
transf-480.png	46.945	PNG image	2016.01.09 16:3...	0644
transf-768.png	112.031	PNG image	2016.01.09 16:3...	0644
transf.png	622.447	PNG image	2015.12.17 15:2...	0644

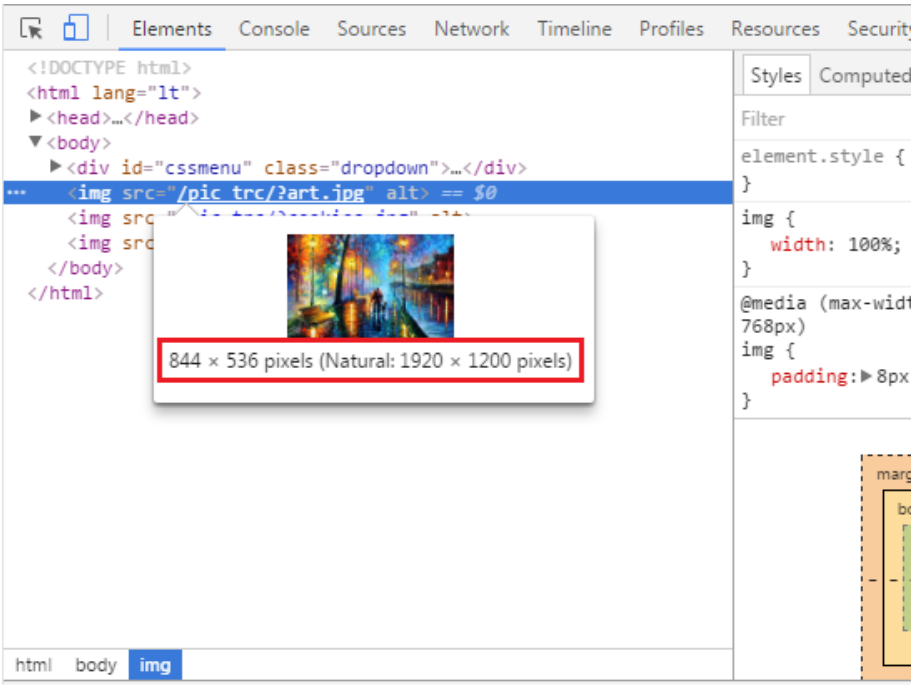
2.12. pav. Paveikslėlių sąrašas ir PHP funkcija, parenkanti reikiamo dydžio paveikslėlį

Kreipiantis į /pic_trc/ aplanką, pirmiausia yra kraunamas index.php failas. Jame aprašytos taisyklės, kurios tikrina slapukų duomenis ir parenka tinkamą paveikslėlį naršyklės ekranui (žr. 2.13 pav.).

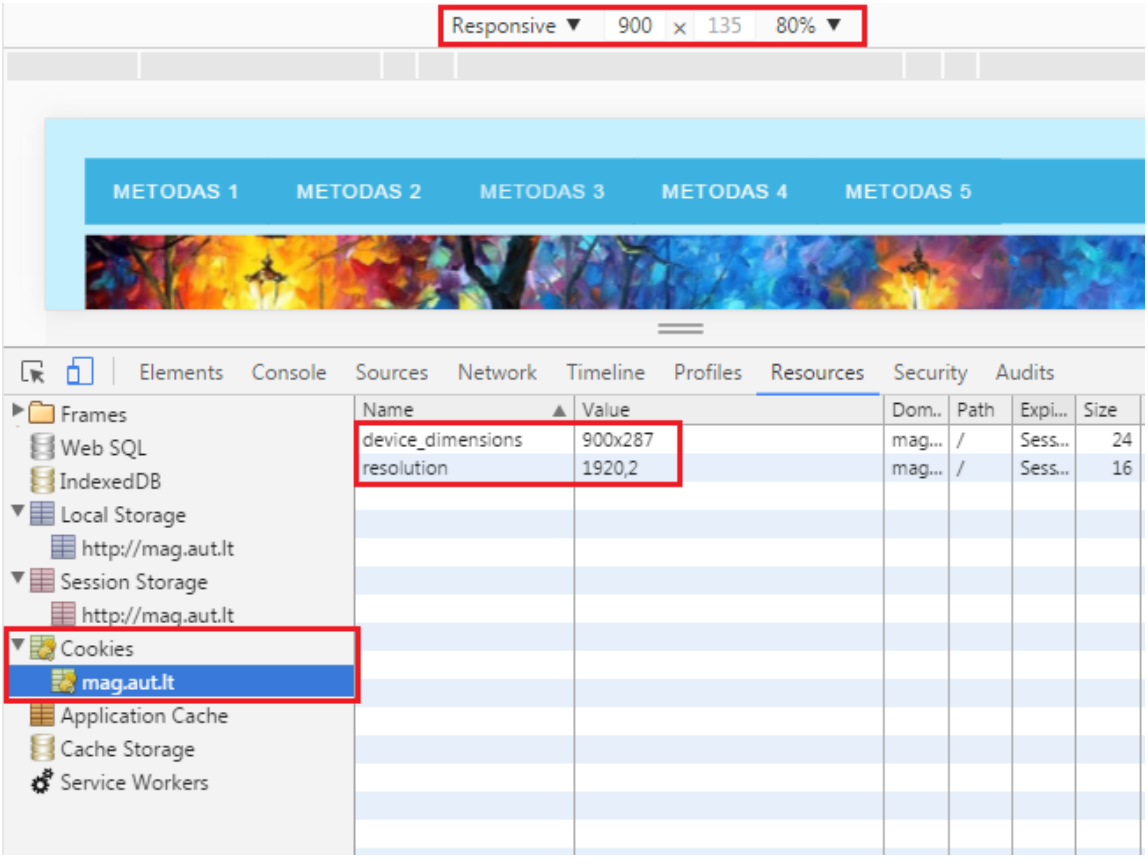
```php
<?php
  $device_width = 0;
  $device_height = 0;
  $file = $_SERVER['QUERY_STRING'];

  if (file_exists($file)) {

    // Read the device viewport dimensions
    if (isset($_COOKIE['device_dimensions'])) {
      $dimensions = explode('x', $_COOKIE['device_dimensions']);
      if (count($dimensions) == 2) {
        $device_width = intval($dimensions[0]);
        $device_height = intval($dimensions[1]);
      }
    }

    if ($device_width > 0) {

      $fileext = pathinfo($file, PATHINFO_EXTENSION);

      // Low resolution image
      if ($device_width <= 480) {
        $output_file = substr_replace($file, '-480', -strlen($fileext) - 1, 0);
      }

      // Low resolution image
      else if ($device_width >= 480 and $device_width <= 768) {
        $output_file = substr_replace($file, '-768', -strlen($fileext) - 1, 0);
      }

      // Low resolution image
      else if ($device_width >= 768 and $device_width <= 1024) {
        $output_file = substr_replace($file, '-1024', -strlen($fileext) - 1, 0);
      }

      // Low resolution image
      else if ($device_width >= 1024 and $device_width <= 1366) {
        $output_file = substr_replace($file, '-1366', -strlen($fileext) - 1, 0);
      }

      // check the file exists
      if (isset($output_file) && file_exists($output_file)) {
        $file = $output_file;
      }
    }

    // return the file;
    readfile($file);
  }
?>
```

2.13. pav. Sąlygos, pagal kurias (priklausomai nuo ekrano dydžio) parenkamas paveikslėlis

COOKIES metodas, taip pat kaip ir anksčiau pristatytas PHP metodas, saugo naršyklės ekrano parametrus, todėl sumažinus naršyklės langą ir jį atnaujinus, parsiunčiamas tas pats failas tol, kol nepakeičiami slapukų duomenys (žr. 2.14 pav., 2.15 pav.).



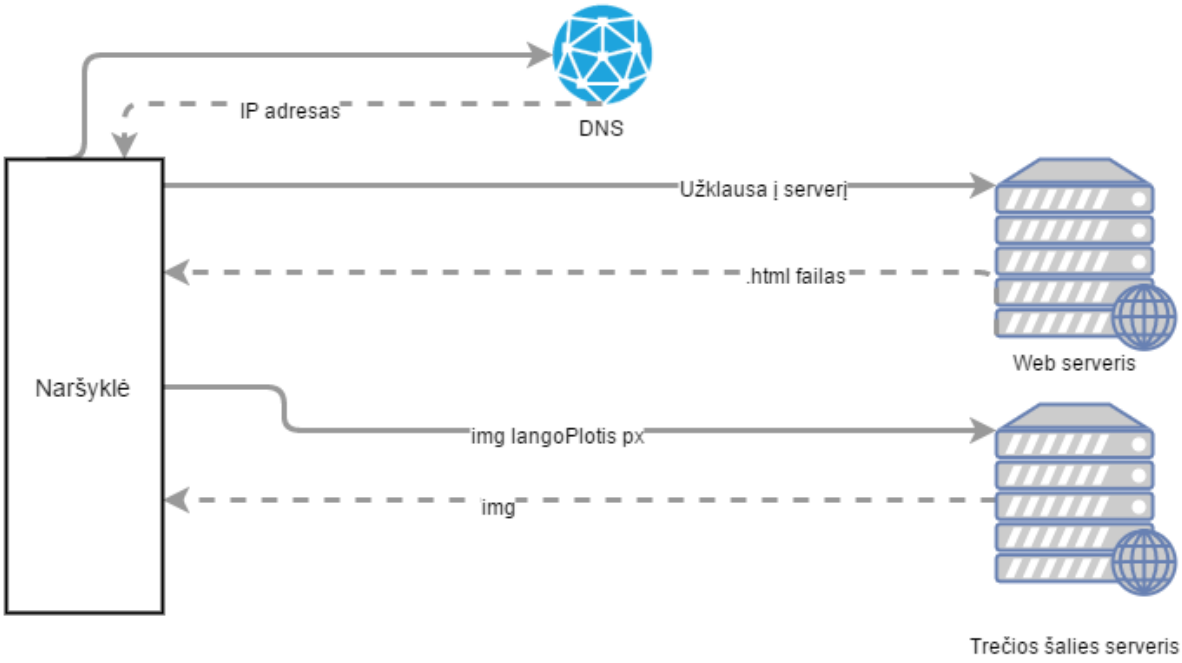
2.14. pav. Mažesniame naršyklės lange užkraunamas paveikslėlis pagal slapukų duomenis



2.15. pav. Slapukų saugoma informacija

1.8.4. RESRC metodas

Literatūros analizėje aprašytas CDN turinio pateikimo metodas, kuris naudojamas optimizuojant serverio resursus ir turinio pristatymo spartą. Šiuo metu galima rasti daug šios paslaugos teikėjų. Kai kurie iš jų siūlo specializuotus paveikslėlių pristatymo tinklus, kurie ne tik saugo paveikslėlius, tačiau juos paruošia įvairaus dydžio ekranams (žr. 2.16 pav.) [28].



2.16. pav. Paveikslėlio užklausa iš kito serverio ir jo atsiuntimas

Tiriamas metodas veikia pirmą kartą nuskaitęs failą iš pagrindinio serverio, šiuo atveju iš mag.aut.lt. HTML dokumente įterpiama „JavaScript“ eilutė (pagal paslaugos teikėjo dokumentaciją [29]) ir nurodomas didelės raiškos paveikslėlio šaltinis (žr. 2.17 pav.).

Šis metodas palaikomas visų modernių naršyklių (nuo Internet Explorer 6 versijos), kuriose veikia „Javascript“.

```
<!DOCTYPE html>
<html lang="lt">
▼<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="styles.css">
    <script src="http://code.jquery.com/jquery-latest.min.js" type="text/
    javascript"></script>
    <style type="text/css"></style>
    <script src="script.js"></script>
    <script src="//use.resrc.it/0.9"></script>
    <script>
            resrc.ready(function() {
            resrc.run();
            });
        </script>
    <title>Magistrinis IMF</title>
  </head>
▼<body>
  ▶<div id="cssmenu" class="dropdown">…</div>
    <img data-src="http://mag.aut.lt/pic/1920/art.jpg" alt class="resrc" src=
    "http://app.resrc.it/s=w1390,pd1/o=85/http://mag.aut.lt/pic/1920/art.jpg"> ==
    <img data-src="http://mag.aut.lt/pic/1920/cookies.jpg" alt class="resrc" s#@=
    "http://app.resrc.it/s=w1390,pd1/o=85/http://mag.aut.lt/pic/1920/cookies.jpg">
    <img data-src="http://mag.aut.lt/pic/1920/transf.png" alt class="resrc" src=
    "http://app.resrc.it/s=w1390,pd1/o=85/http://mag.aut.lt/pic/1920/transf.png">
  </body>
</html>
```

2.17. pav. HTML dokumente naudojamos žymos

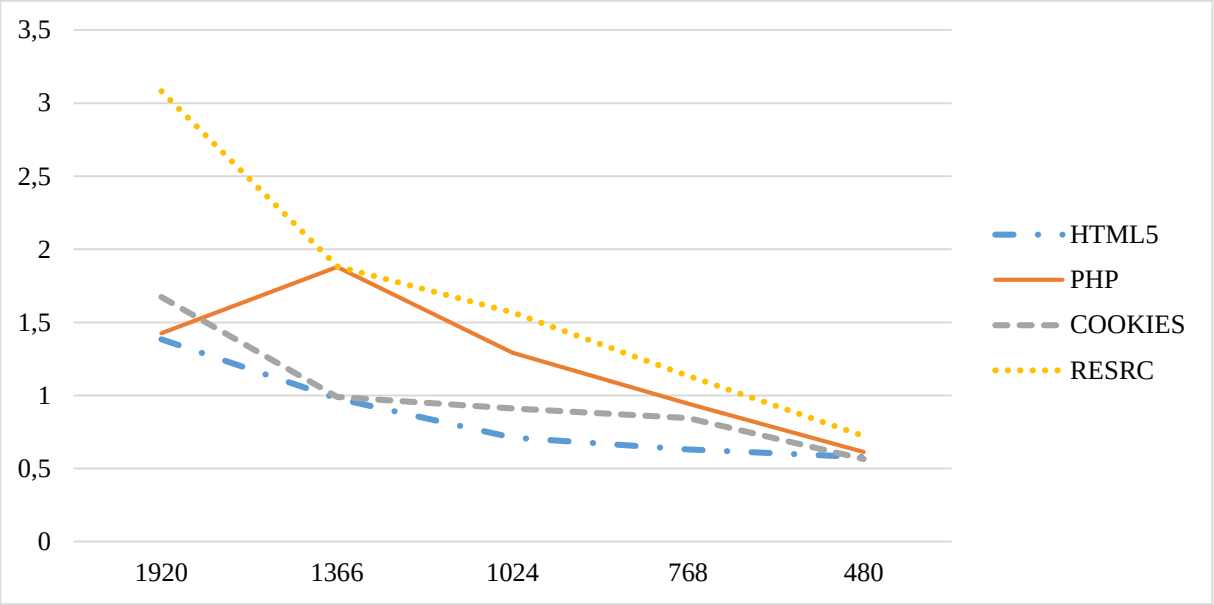
CDN gavęs paveikslėlį jį apdoroja ir grąžina reikiamo dydžio vaizdą (žr. 2.18 pav.). Visi paveikslėliai optimizuojami 85 proc. kokybe.

2.18. pav. Vaizdo grąžinimas pagal ekrano plotį

3. METODŲ TYRIMAS

1.9. Grafinių elementų užkrovimo laikas

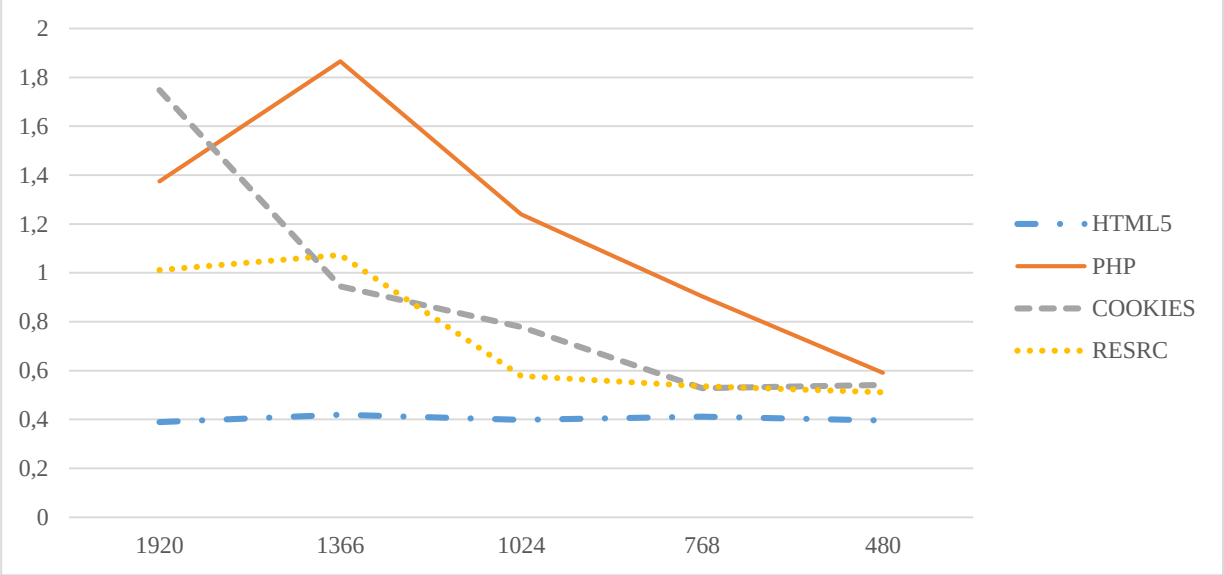
Tiriant metodus Chrome naršyklėje nustatytas rezoliucijos taško santykis – 1. Gauti rezultatai (žr. 3.1 pav.) lyginami pagal parsiunčiamų vaizdų dydį bei užkrovimo laiką. Tyrimas atliekamas nesaugant informacijos naršyklėje (angl. *disable cache*), 50 kartų kartojant užklausas. Remiantis 1.6 skyriuje aprašytais duomenimis, metodai tiriami 1920, 1366, 1024, 768 ir 480 taškų pločio naršyklėse.



3.1. pav. Grafinių elementų užkrovimo laikas (s)

Geriausius rezultatus pasiekė HTML5 metodas. Jo užkrovimo laikas buvo mažiausias, 4 iš 5 tiriamų ekranų dydžių, kurie buvo parinkti 1.6 skyriuje, užsikrovė greičiausiai. Panašius rezultatus pasiekė metodas su slapukais.

Toliau tyrimas kartojamas atnaujinant naršyklės langą su informacijos saugojimu (angl. *caching*). Tyrime pateikti duomenys remiasi 50 bandymų, jų rezultatams išvedus vidurkį (žr. 3.2. pav.).



3.2. pav. Tinklalapio užkrovimo laikas turinį užkraunant pakartotinai (s)

Vertinant rezultatus 1920px pločio ekrane HTML5 metodas turinį užkrovė greičiausiai (0,389 s). Lyginant su pirmu užkrovimu, COOKIES ir PHP metodai turi panašius rezultatus (1,748 ir 1,374 s atitinkamai). RESRC metodas turi didžiausią absoliutųjį pokytį palyginus su pirmu užkrovimo laiku.

1.10. Užkraunamų paveikslėlių dydis

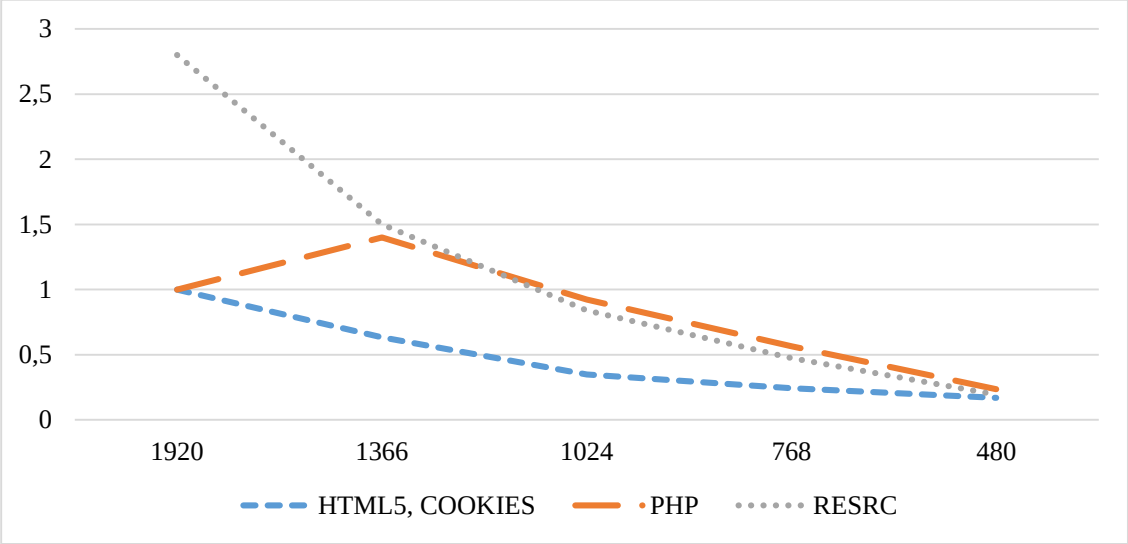
Lyginant parsiunčiamų failų dydį matyti, jog HTML5 ir COOKIES naudoja tuos pačius paveikslėlius, todėl jų dydis sutampa.

Užkrovimo metu tiriamas ir užkraunamų vaizdų dydis, kadangi dideli failai išnaudoja didelį serverių srautą. Tai taip pat turi įtakos užkrovimo greičiui. Žemiau pateiktame grafike (žr. 3.4 pav.) matyti, jog esant 1920px pločio ekranui HTML5, PHP ir COOKIES metodas pateikia tokio pačio dydžio failus. Visų metodų atveju tai normalu, kadangi naudojami tie patys vaizdai, tačiau parenkami skirtingais metodais. PHP metodo veikimo atveju yra nurodomas didžiausias paveikslėlis, kuris mažesniems ekranams mažinamas naudojant PHP funkciją pagal nurodytą dydžių masyvą (žr. 3.3. pav.).

```
/* CONFIG ---------------------------------------------------------

$resolutions    = array(1920, 1366, 1024, 768, 480)  // the resolu
$cache_path     = "ai-cache"; // where to store the generated re-s
$jpg_quality    = 80; // the quality of any generated JPGs on a sc
$sharpen        = TRUE; // Shrinking images can blur details, perf
$watch_cache    = TRUE; // check that the adapted image isn't stal
$browser_cache = 200; // How long the BROWSER cache should last (
```

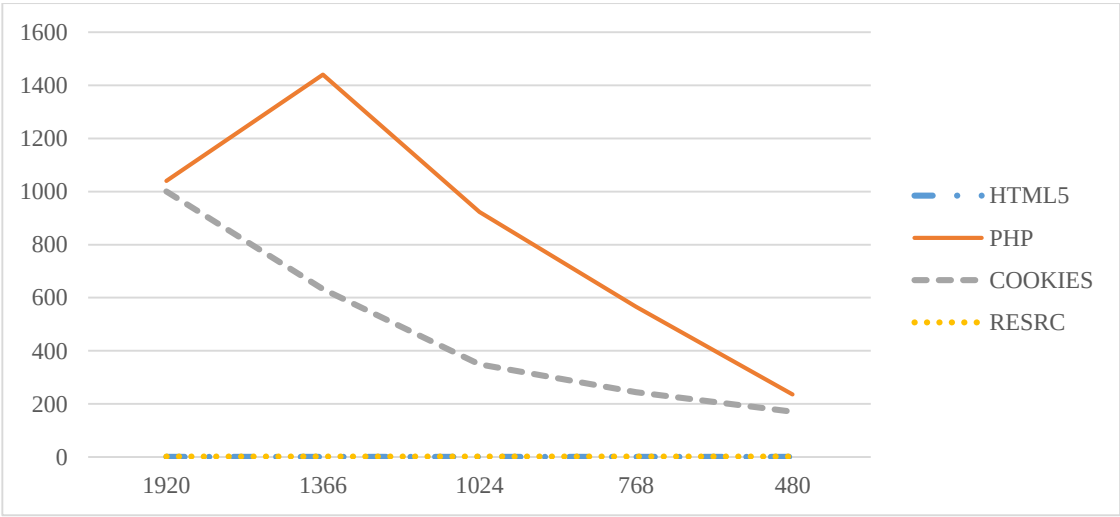
3.3. pav. PHP metodo konfigūravimas

Didžiausias failų dydis, užkrautas iš trečiosios šalies serverio (RESRC metodas), esant 1920px pločio ekranui – 2,8 MB (žr. 3.4. pav.). Naršyklės lange, kurios plotis 1366px, didžiausio srauto pareikalavo RESRC metodas – 1,5 MB, tačiau tai ženkliai mažesnis atotrūkis lyginant su kitais metodais, ypač PHP metodu, kurio vaizdai užėmė 1,4 MB. Ištirta, kad pastarojo metodo vaizdų dydis 1366px pločio ekrane reikalauja didesnio srauto nei 1920px pločio ekrane. Tai galima paaiškinti neefektyviu paveikslėlių optimizavimu juos mažinant reikiamam dydžiui, kas buvo atlikta mažinant vaizdus kitiems metodams.



3.4. pav. Vaizdų dydis (MB) pagal ekrano plotį

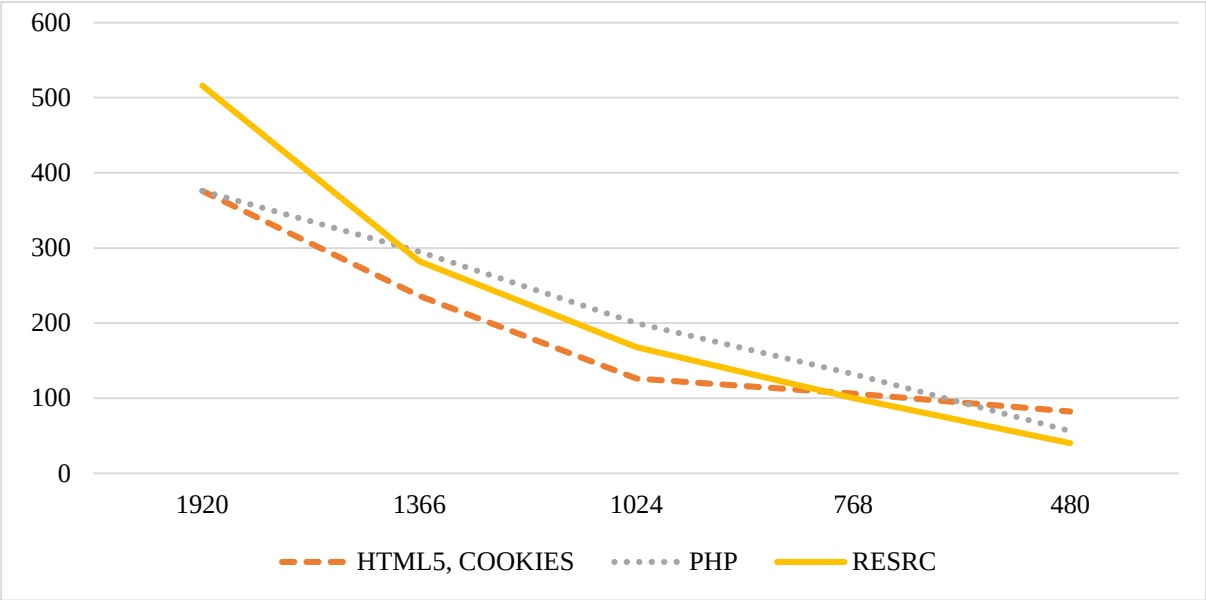
Atlikus tyrimą su metodų saugomais duomenimis matyti, kad PHP ir COOKIES metodai neišsaugo duomenų ir kiekvieno pakartotinio užkrovimo metu visas grafinis turinys yra parsiunčiamas iš naujo (žr. 3.5 pav.). Tai yra viena iš priežasčių, kodėl šių metodų rezultatai buvo prasčiausi.



3.5. pav. Antrą kartą užkraunamų duomenų dydis (KB)

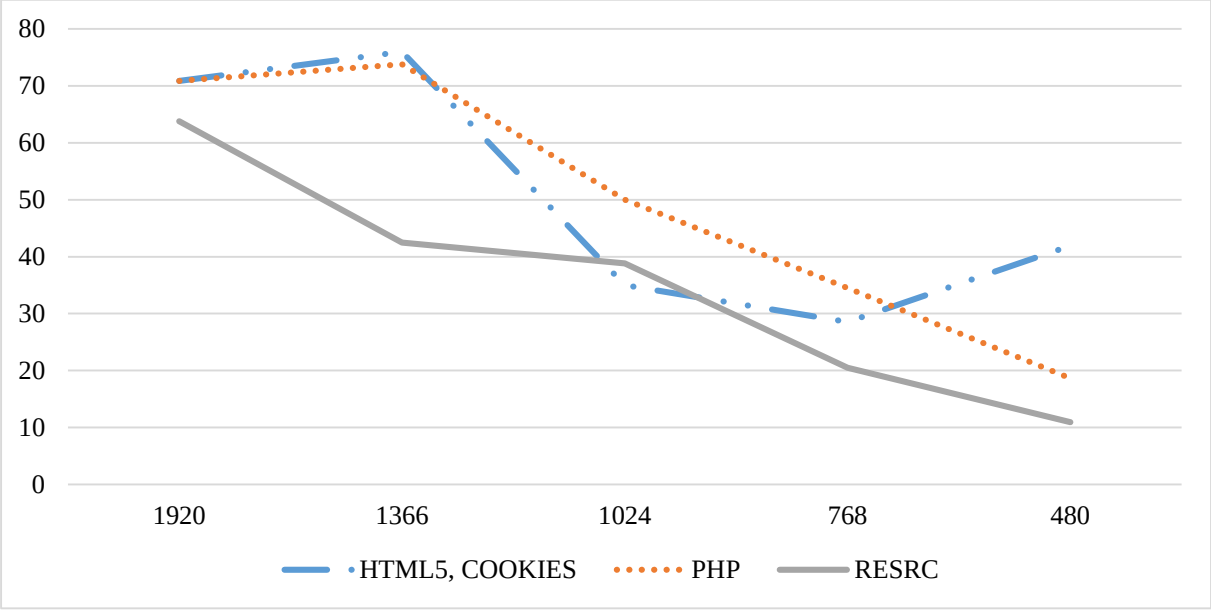
1.10.1. Skirtingų formatų paveikslėlių užkrovimas

Lyginami metodai netik naudoja skirtingus vaizdų parinkimo būdus, tačiau parsiunčia skirtingų dydžių failus. Kaip buvo minėta, tyrimui naudoti skirtingi formatai: jpg ir .png (žr. 3.6 pav.).



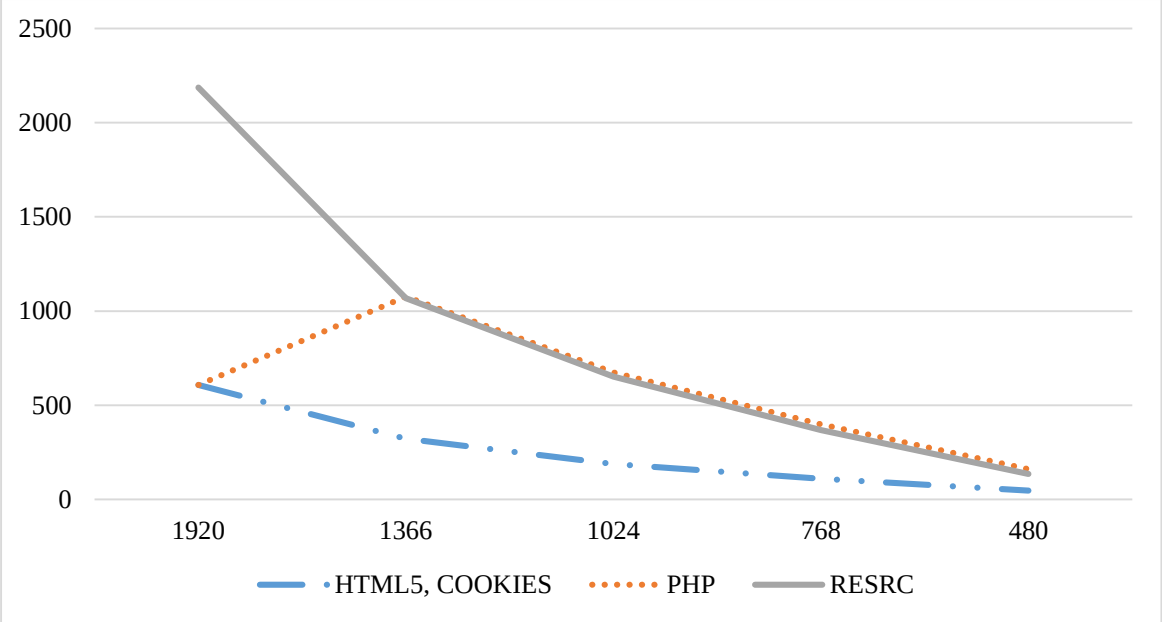
3.6. pav. Vaizdo art.jpg parsiunčiamo failo dydis skirtingais metodais (KB)

RESRC metodas efektyviai sumažina .jpg formato paveikslėlius, ypač mažesniems ekranams (žr. 3.7 pav.). Ekranui, kurio plotis 480px, parsiunčiamo paveikslėlio dydis buvo 40 KB.



3.7. pav. Vaizdo cookies.jpg parsiunčiamo failo dydis taikant skirtingus metodus

Didžiausią skirtumą galima pastebėti parsiunčiant .png formato failą. Tai rodo neefektyvų RESRC ir PHP metodų .png formato failų optimizavimą (žr. 3.8 pav.).



3.8. pav. Vaizdo transf.png parsiunčiamo failo dydis taikant skirtingus metodus

1.11. Koeficientų priskyrimas pagal ekrano didžių pasiskirstymą

Norint išrinkti greičiausiai paveikslėlius pristatantį metodą reikia išvesti metodų rezultatų vidurkį. Metodų efektyvumas skirtingų dydžių ekranuose skiriasi, todėl aritmetinis vidurkis pateiktų rezultatą, kuris realiame pasaulyje nepasiteisintų. T. y. greičiausias metodas nebūtinai būtų tinkamiausias 1366px pločio ekranuose, nors statistiškai toks ekranas, naršant žiniatinklyje, naudojamas dažniausiai. Didžiausias prioritetas turėtų būti teikiamas dažniausiai naudojamų ekranų dydžiams. Objektyviam metodų įvertinimui, skaičiuojant vidurkį, galima priskirti koeficientus pagal ekranų panaudojimą.

Koeficientas apskaičiuojamas pagal ekranų panaudojimo dažnumą (žr. 1.8 pav.). Naudojami tik tyrimui parinkti ekrano dydžių rezultatai. Koeficientų suma lygi – 1 (žr. 3.1 lentelė).

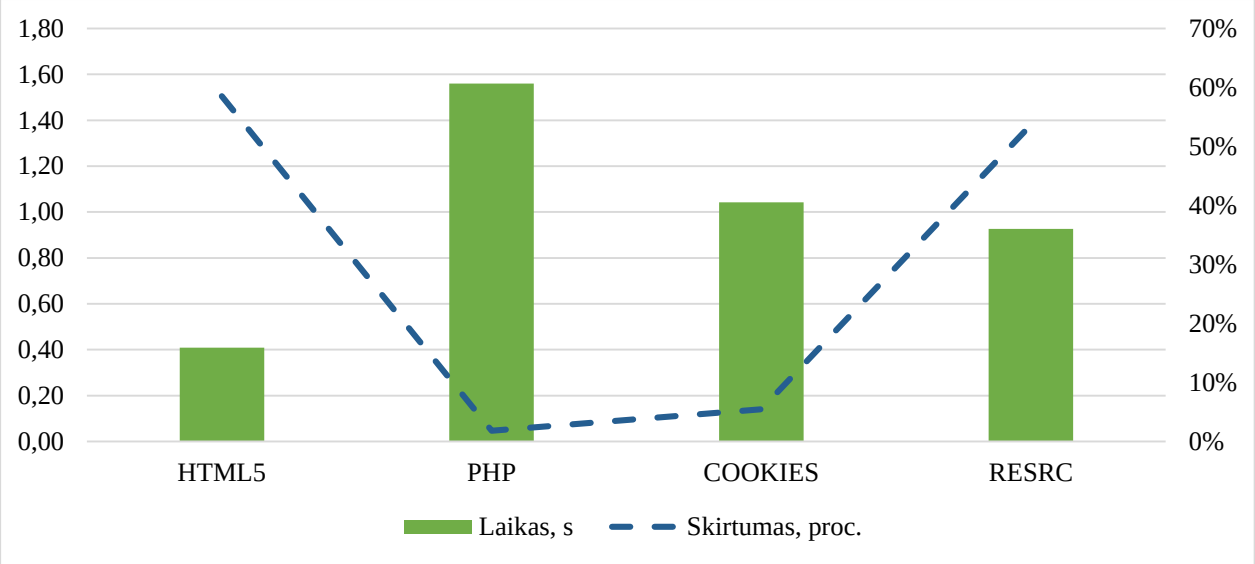
3.1. lentelė. Priskiriamų koeficientų skaičiavimas

Ekrano plotis, px	Panaudojimas	Simbolis	Apskaičiavimas	Koeficientas
1366	29,88	A	A/F	0,53
1920	11,98	B	B/F	0,21
1024	8,55	C	C/F	0,15
768	5,79	D	D/F	0,1
480	0,47	E	E/F	0,01
Suma:	**56,67**	**F**	**Suma:**	**1**

Pagal šiuos gautus koeficientus skaičiuojamas vidurkis. Kiekvienas ekrano dydžio rezultatas dauginamas iš koeficiento. Geriausią rezultatą pasiekė HTML5 metodas, kurio vidurkis siekė 0,98 s (žr. 3.9 pav.). Ilgiausią užkrovimo laiką turi RESRC metodas, kurio vidurkis – 2 s.

3.9. pav. Metodų rezultatų vidurkis pritaikius koeficientus

Pritaikant tuos pačius koeficientus ir gavus laiko vidurkius, užkraunant grafinius elementus pakartotinai su išsaugota tinklalapio informacija, matomi skirtumai tarp pirmo ir pakartotinių užkrovimų (žr. 3.10 pav.). Šie rezultatai rodo, kaip greitai elementai užkraunami tinklalapyje jį atvaizduojant pakartotinai. Iš duomenų matyti, kad pakartotinai užkrovus tinklalapį HTML5 metodas turinį atvaizdavo per 0,41 s. Tai yra 59 proc. greičiau nei pirmo užkrovimo metu. Pakartotinai užkraunant turinį ilgiausiai krovė PHP metodas (skirtingai nei tiriant pirminį užsikrovimą, kurio metu RESRC metodas krovė ilgiausiai). Jis turinį atvaizdavo per 1,56 s ir tai buvo 2 proc. greičiau nei pirmo užkrovimo metu. COOKIES metodas pakartotinai užkraunant turinį neparodė ženkliai geresnių rezultatų, nei pirmo užkrovimo metu – 1,04 s, t. y. 5 proc. mažesnis laikas. Pakartotinai užkraunant tinklalapį pagerėjo RESRC metodo rezultatai (nepaisant didelio duomenų pasiskirstymo (imties dispersija – 0,28)). Šio metodo užkrovimo laiko vidurkis – 0,93 s. ir, lyginant su pirmu užkrovimu, kuomet užkrovimo laikas buvo ilgiausias, tai sudaro 54 proc. trumpesnį laiką.



3.10. pav. Grafinių elementų užkrovimas pakartotinai

Skaičiuojant paveikslėlių dydžių koreliavimą su užkrovimo laiku, gaunamas 0,98 rodiklis (pirmo užkrovimo metu), kuris rodo glaudų ryšį tarp parsiunčiamų failų dydžio ir laiko (žr. 3.2 lentelė). Todėl galima teigti, kad efektyviai sumažinus paveikslėlių dydį galima pasiekti greitesnį jų atvaizdavimą.

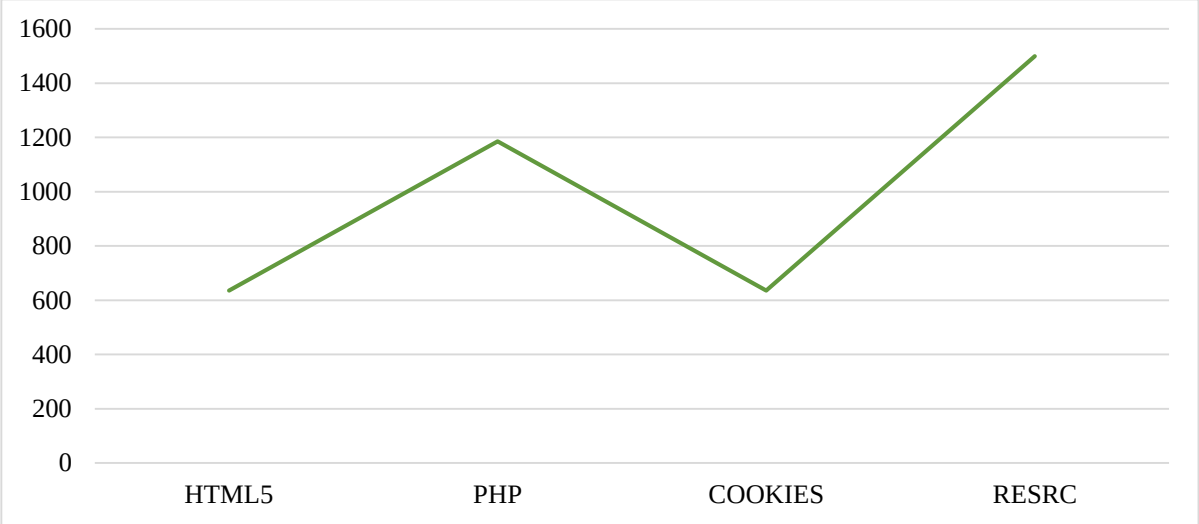
3.2. lentelė. Failų dydžio ir užkrovimo laiko koreliacija

	Paveikslai, KB	**Užkrovimas, s**		**Paveikslai, KB**	**Užkrovimas, s**
Pirmas užkrovimas	1054,9	1,384	Pakartotinas užkrovimas	0,505	0,3892
	634	0,9786		0,505	0,4188
	349	0,7124		0,505	0,3982
	244,5	0,63		0,505	0,4114
	170,3	0,5778		0,505	0,396
	1054,9	1,426		1040	1,374
	1444,8	1,878		1440	1,866
	924	1,292		923	1,24
	565,5	0,9432		566	0,9046
	236	0,6136		236	0,5914
	1054,9	1,674		1000	1,748
	634	0,9894		633	0,9442
	349	0,9108		349	0,7784
	244,5	0,845		244	0,5284
	170,3	0,5648		170	0,5408
	2766,8	3,082		1,4	1,012
	1393,5	1,884		1,4	1,0736
	858,8	1,567		1,4	0,5778
	488,5	1,134		1,4	0,5364
	185,9	0,722		1,4	0,5112
Koreliacija	**0,98**		**Koreliacija**	**0,88**	

Pakartotinai užkraunant tinklalapį failų dydis turi mažesnę įtaką užkrovimo greičiui (koreliacijos rodiklis sudaro 0,88), nes HTML5 ir RESRC metodai atvaizduoja jau išsaugotus paveikslėlius naršyklėje.

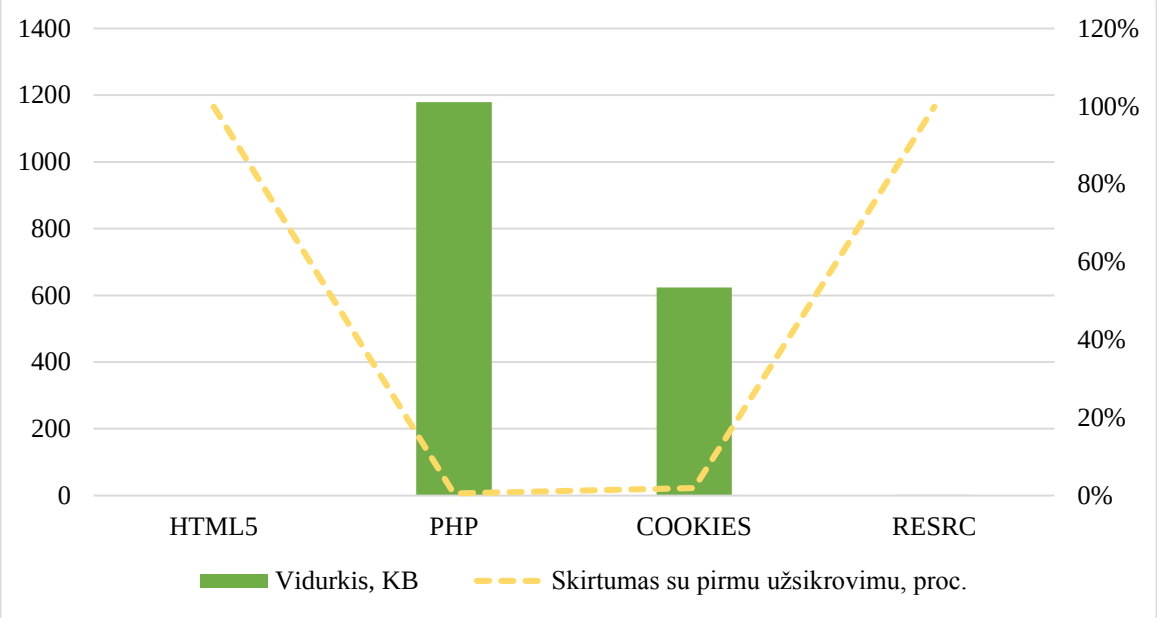
1.12. Koeficientų priskyrimas pralaidumo vertinimui

Lyginant metodų pralaidumo rezultatus yra išvedamas bendras metodo rodiklis, kuriam priskiriami ankstesniame skyriuje išvesti koeficientai (žr. 3.1 lentelė).



3.11. pav. Paveikslėlių dydžių vidurkis pagal koeficientus (KB)

Pritaikius vidurkio skaičiavimą pastebima, jog HTML5 ir COOKIES metodų paveikslėliai pareikalavo tokio pačio duomenų srauto – 636 KB (žr. 3.11 pav.). Didžiausio srauto pareikalavo RESRC metodas. Jo parsiunčiamų paveikslėlių dydžio vidurkis sudarė 1499 KB.



3.12. pav. Parsiųstų paveikslėlių dydžių vidurkis kartojant užkrovimą

HTML5 ir RESRC metodai pakartotinio užkrovimo metu paveikslėlių nebesiunčia (jie yra išsaugomi naršyklėje). Užkraunamas tik .html failas, kuris sudaro 0,51 ir 1,4 KB atitinkamai (žr. 3.12 pav.). PHP metodas kiekvieno užkrovimo metu siunčia tuos pačius failus, kurių vidurkis sudaro 1179 KB. COOKIES metodas taip pat atsiunčia visus jam nurodytus vaizdus. Atsiųstų vaizdų dydis – 624 KB ir tai yra 1,9 proc. mažiau nei pirmo užkrovimo metu parsiunčiamų failų dydis.

1.13. Rezultatai

Atlikus duomenų analizę galima išskirti du grafinių elementų atvaizdavimo atvejus: pirmo užkrovimo momentas ir pakartotinis užkrovimas.

Iš lentelės matyti, kad HTML5 ir COOKIES metodai atsiunčia tą patį duomenų kiekį, tačiau HTML5 užkrauna turinį 0,12 s arba 10,9 proc. greičiau. Trūkumas – HTML metodas yra naujausias iš visų tirtų metodų ir nei viena Internet Explorer naršyklė jo pilnai nepalaiko, tačiau atvaizduoja žymoje nurodytą numatytąjį paveikslėlį.

3.3. lentelė. Pirmo užkrovimo metodų duomenų lentelė

Požymis	**HTML5**	**PHP**	**COOKIES**	**RESRC**
Užkrovimo laikas, s	0,98	1,59	1,10	2,00
Pralaidumas, KB	636	1185	636	1499
Kaina	Nemokama	Nemokama	Nemokama	13 Eur/mėn
Palaikymas	<picture> žyma	„media querie“	„media querie“	„JavaScript“

Atsiunčiamų duomenų išsaugojimas gali ne tik paspartinti tinklalapio užkrovimą, bet ir sumažinti serverio resursus. Tyrimo rezultatai atskleidė, kad HTML5 metodas iš serverio parsiuntė vidutiniškai 0,51 KB duomenų (žr. 3.4 lentelė). RESRC metodas iš paslaugos teikėjo serverio parsiuntė 1,4 KB duomenų, kas yra artima HTML5 metodo rezultatui. Tačiau RESRC metodas vaizdus užkrovė per 0,93 s, tuo tarpu HTML5 metodas vaizdus pateikė per 0,41 s arba 55,9 proc. greičiau.

3.4. lentelė. Pasikartojančio užkrovimo duomenų lentelė

Požymis	**HTML5**	**PHP**	**COOKIES**	**RESRC**
Užkrovimo laikas, s	0,41	1,56	1,04	0,93
Pralaidumas, KB	0,51	1179,01	623,94	1,40
Kaina	Nemokama	Nemokama	Nemokama	13 Eur/mėn
Palaikymas	<picture> žyma	„media querie“	„media querie“	„JavaScript“

PHP metodas pakartotinai užkrovė tinklalapį per panašų laiką (1,56 s) kaip pirmo krovimo metu, taip yra dėl didelio duomenų kiekio (1179 KB). Metodai PHP ir COOKIES, neišsaugantys duomenų naršyklėje, pareikalauja didelio duomenų pralaidumo iš serverio, kuomet tinklalapis užklausiamas iš kelių pakartotinai apsilankiusių žiniatinklio vartotojų.

4. IŠVADOS

1) Atlikus literatūros analizę paaiškėjo, kad 63,7 proc. naršyklėje užkraunamo turinio sudaro grafiniai elementai. Tai rodo, kad šio tipo failų optimizavimas gali paspartinti tinklalapio užkrovimą. Taip pat iš pasaulinės statistikos duomenų matyti, kad daugėja vartotojų, kurie turinį peržiūri dideliuose ekranuose. Tad greitam tinklalapio užkrovimui ypač aktualus didelės rezoliucijos grafinių elementų užkrovimo greitis.

2) Tinklalapio pilno užkrovimo laikas yra pagrindinis metodų vertinimo kriterijus. Metodai papildomai vertinami pagal parsiunčiamų duomenų srautą, metodo eksploatavimo kainą bei palaikymą šiuo metu naudojamose naršyklėse. Tiriant galimus grafinių elementų atvaizdavimo metodus naršyklėje, rasti keturi skirtingi metodai (HTML5, PHP, COOKIES ir RESRC), kurie parenka reikiamo dydžio vaizdą pagal ekrano dydį bei taškų intensyvumą. Tyrimo objektui numatomas .jpg ir .png failų atvaizdavimas įvairaus dydžio naršyklių languose.

3) Realizavus prototipą su skirtingais paveikslėlių parinkimo metodais (HTML5, COOKIES, PHP ir RESRC) išsiaiškinta, kad tyrimui tinkamiausias Google DevTools įrankis. Šis įrankis užfiksuoja užklausos išsiuntimo, serverio atsakymo ir failų užkrovimo laiką bei sunaudotą duomenų srautą.

4) Lyginant užkrovimo greitį, visuose ekrano dydžiuose HTML5 rodiklis buvo mažiausias – 0,98 s užkraunant pirmą kartą ir 0,41 s užkraunant turinį pakartotinai. Ilgiausias krovimo laikas (kraunant pirmą kartą) nustatytas taikant RESRC metodą (2 s). Pakartotinio užkrovimo metu prasčiausiai pasirodė PHP metodas, kurio krovimosi laiko vidurkis buvo 1,56 s. Mažiausią duomenų srautą pirmo krovimo metu sunaudoja HTML5 ir COOKIES metodai – 636 KB. Daugiausia – RESRC metodas (1499 KB). Antro užsikrovimo metu HTML5 ir RESRC metodai duomenis išsaugodavo naršyklėje, todėl duomenų sunaudojimas atitinkamai sumažėjo iki 0,51 ir 1,4 KB. Kadangi PHP ir COOKIES metodai duomenų naršyklėje neišsaugojo, todėl iš serverio parsiuntė 1179 ir 624 KB dydžio failus. Pirmo užkrovimo metu tiriant ryšį tarp parsiųstų failų dydžio ir užkrovimo laiko, gautas 0,98 koreliacijos koeficientas. Šis rodiklis rodo optimizuotų paveikslėlių svarbą pirmo užkrovimo metu. Pakartotinio užkrovimo metu šis ryšys tapo mažiau glaudus (0,88) dėl duomenų išsaugojimo naršyklėje (HTML5 ir RESRC metodai).

5. LITERATŪRA

[1] S. C. Colleen Van Lent, „Design of Complex Websites,“ [Tinkle]. Available: https://www.si.umich.edu/programs/class/winter-2014/design-complex-websites-0. [Kreiptasi 20 sausis 2015].

[2] D. Comer, Computer Networks and Internets, 99 p., Prentice Hall, 2008.

[3] S. K. A. Gessler, „PDAs as mobile WWW browsers,“ [Tinkle]. Available: http://www.sciencedirect.com/science/article/pii/0169755295000936. [Kreiptasi 5 balandis 2016].

[4] Statista, „Average global internet connection speed from 1st quarter 2011 to 4th quarter 2015 (in Mbps),“ [Tinkle]. Available: http://www.statista.com/statistics/204954/average-internet-connection-speed-worldwide/. [Kreiptasi 12 gegužė 2016].

[5] „Internet live stats,“ [Tinkle]. Available: http://www.internetlivestats.com/internet-users/#trend. [Kreiptasi 16 gruodis 2015].

[6] T. Berners-Lee, „Information Management: A Proposal,“ įtraukta *CERN*, 1989.

[7] „HTTP archive,“ [Tinkle]. Available: http://httparchive.org/interesting.php#bytesperpage. [Kreiptasi 27 kovas 2015].

[8] W. W. W. Consortium, „W3C,“ [Tinkle]. Available: http://www.w3.org/TR/CSS21/syndata.html#q10. [Kreiptasi 17 sausis 2015].

[9] P.-P. Koch, „A pixel is not a pixel is not a pixel,“ [Tinkle]. Available: http://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html. [Kreiptasi 28 kovas 2015].

[10] M. Allin, „What Is JavaScript?,“ [Tinkle]. Available: http://media.wiley.com/product_data/excerpt/88/07645790/0764579088.pdf. [Kreiptasi 17 sausis 2015].

[11] T. M. Ö. Ling Liu, įtraukta *Encyclopedia of Database Systems*, 2009, p. 41.

[12] R. K. S. J. S. Erik Nygren, „The Akamai Network: A Platform for High-Performance Internet Applications,“ įtraukta *ACM SIGOPS Operating Systems Review*, 2010.

[13] Google, „Chrome DevTools Overview,“ Google, [Tinkle]. Available: https://developer.chrome.com/devtools. [Kreiptasi 17 balandis 2016].

[14] Z. M. Gillenwater, „Examples of flexible layouts with CSS3 media queries,“ įtraukta *Stunning CSS3*, 2010, p. 320.

[15] G. Stats, „Screen resolution monthly,“ [Tinkle]. Available: http://gs.statcounter.com/#all-resolution-ww-monthly-201411-201511. [Kreiptasi 5 gruodis 2015].

[16] P. LePage, „Responsive Web Design Basics,“ 2015. [Tinkle]. Available: https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/. [Kreiptasi 20 sausis 2015].

[17] M. Developer, „Windows Presentation Foundation Graphics Rendering Overview,“ 29 kovas 2015. [Tinkle]. Available: https://msdn.microsoft.com/fr-fr/library/ms748373(v=vs.85).aspx.

[18] W. W. W. Consortium, „The Picture Element,“ [Tinkle]. Available: http://www.w3.org/html/wg/drafts/html/master/semantics.html#the-picture-element. [Kreiptasi 29 kovas 2015].

[19] M. D. N. a. i. contributors, „<picture> tag,“ [Tinkle]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/picture. [Kreiptasi 8 d. Rugpjūtis 2015].

[20] M. Wilcox, „Adaptive-image,“ [Tinkle]. Available: http://adaptive-images.com/. [Kreiptasi 5 birželis 2015].

[21] K. Clark, „Responsive images using cookies,“ [Tinkle]. Available: http://keithclark.co.uk/articles/responsive-images-using-cookies/. [Kreiptasi 23 rugpjūtis 2015].

[22] S. Alexander, „Choosing A Responsive Image Solution,“ 5 4 2015. [Tinkle]. Available: http://www.smashingmagazine.com/2013/07/08/choosing-a-responsive-image-solution/.

[23] M. D. N. a. i. Contributors, „@media,“ [Tinkle]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS/@media. [Kreiptasi 17 balandis 2016].

[24] S. Global, „Screen resolution monthly,“ [Tinkle]. Available: http://gs.statcounter.com/#all-resolution-ww-monthly-201502-201602. [Kreiptasi 12 kovas 2016].

[25] „Desktop browsers,“ [Tinkle]. Available: http://gs.statcounter.com/#desktop-browser-ww-monthly-200807-201506. [Kreiptasi 17 sausis 2015].

[26] W3Techs, „Usage of image file formats for websites,“ Q-Success, [Tinkle]. Available: http://w3techs.com/technologies/overview/image_format/all. [Kreiptasi 3 balandis 2016].

[27] TinyPNG, „Compress PNG images while preserving transparency,“ TinyPNG, [Tinkle]. Available: https://tinypng.com/. [Kreiptasi 16 sausis 2016].

[28] Nežinoma, „The picture element,“ [Tinkle]. Available: https://html.spec.whatwg.org/multipage/embedded-content.html#the-picture-element. [Kreiptasi 18 vasaris 2016].

[29] E. Thurgood, „ReSRC,“ [Tinkle]. Available: https://www.resrc.it/. [Kreiptasi 7 vasaris 2016].

[30] ReSRC, „Quick Start Guide,“ [Tinkle]. Available: https://www.resrc.it/docs. [Kreiptasi 11 kovas 2016].