

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Tomas Abrutis

**Lygiagretaus programavimo metodai
reklamų apsikeitimo sistemoje**

Magistro darbas

Darbo vadovas
Dr. Romas Marcinkevičius

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Tomas Abrutis

**Lygiagretaus programavimo metodai
reklamų apsikeitimo sistemoje**

Magistro darbas

Recenzentas

Dr. A. Lenkevičius

2010-05

Darbo vadovas

Dr. Romas Marcinkevičius

2010-05

Atliko

IFM – 4/2 gr. stud.

Tomas Abrutis

2010-05

Kaunas, 2010

SUMMARY

Ad management system is developed for place ads on connected sites, for easier, more flexible and faster management. To build this application we are using concurrency methods. They can ensure more stable and faster running.

We using two concurrency methods, and two sequential, to compare them, how they react at different sites number, and to different ads number.

With concurrency methods we get much more better results, when we talking about time. If we have ten sites, and ten ads, sequential methods is five time slower. If we increase ad number or site number this different will be bigger.

At conclusion, we made a decision, that if we want to develop flexible, fast, and more able to work with bigger numbers of sites and ads system, we need use concurrency methods. If sream_select method is to complex, cURL library is good decision.

Turinys:

1	Įvadas.....	6
2	Reklamos valdymo sistemų ypatumai ir naudojimas	8
2.1	Reklamos valdymo sistemų svarba	8
2.2	Reklamų tipai ir reklamos veikimas.....	10
2.3	Reklamos valdymo sistemų rinka ir vartotojai.....	10
3	Lygiagretaus programavimo metodų panaudojimo galimybės valdant reklamos publikavimo serverius.....	12
3.1	Lygiagretaus programavimo metodų naudojimo plusai.....	13
3.2	Lygiagretaus programavimo metodų naudojimo minusai.....	13
4	Centralizuotos reklamos valdymo sistemos aprašymas	14
5	Centralizuotos reklamos valdymo sistemos funkcijos ir galimybė pritaikyti lygiagrečių programavimą	16
5.1	Centralizuotos reklamos valdymo sistemos funkcijos, kurios veikia naudojant lygiagrečius metodus	17
5.2	Centralizuotos reklamos valdymo sistemos funkcijos, kurias galima pritaikyti naudoti lygiagrečius metodus	18
6	Lygiagretaus programavimo metodai, naudojami analizuojant reklamos valdymo sistemą	19
6.1	CURL biblioteka	19
6.2	Stream_select funkcijos panaudojimas	20
6.3	Fopen funkcijos panaudojimas.....	21
6.4	File_get_contents funkcijos panaudojimas	21
7	Pasirinkta tyrimo aplinka.....	23
8	Atliekamo tyrimo rezultatai ir rezultatų analizė.....	25
8.1	Laiko pokytis priklausomai nuo svetainių skaičiaus.....	25
8.2	Laiko pokytis priklausomai nuo reklamų skaičiaus	31
8.3	Bendras metodų įvertinimas.....	35

9	Išvados.....	36
10	Literatūra.....	37
11	Terminų ir santrumpų žodynas	39
12	Priedai	41
12.1	Skirtingų metodų įtaka laikui svetainių patikrinimo metu	41
12.2	Skirtingų metodų įtaka laikui reklamų perkėlimo metu	42
12.3	Skirtingų metodų įtaka laikui reklamų statistikos surinkimo metu	43
12.4	Skirtingų metodų įtaka laikui reklamų perkėlimo metu, kai kinta reklamų skaičius 44	
12.5	Skirtingų metodų įtaka laikui statistikos surinkimo metu, kai kinta reklamų skaičius 45	

1 Įvadas

Reklamos valdymas, viena iš pagrindinių puslapių administravimo dalių, tiesiogiai nešanti pajamas ir reikalaujanti išskirtinio dėmesio. Ši administravimo dalis nėra sudėtinga valdant tik vieną ar dvi svetaines, tačiau tampa komplikuoja, kai vienai firmai priklauso keletas ir daugiau internetinių projektų.

Rinkoje jau yra ne viena reklamos valdymo sistema, pradedant mokamomis sistemomis, ir baigiant atvirojo kodo reklamos valdymo aplinkomis. Taipogi ir gerai žinoma „Google“ reklamos valdymo sistema.

Daugumą reklamos valdymo sistemų sieja vienas valdymo principas. Reklamos talpinamos viename serveryje, kuris atsakingas už jų talpinimą, valdymą, statistikos rinkimą. Tokia architektūra suteikia galimybes paprastesniam valdymui, lengviau sekti statistiką. Tačiau yra ir neigiamų veiksnių. Reikia atskiro reklamos serverio, priklausomai nuo projektų skaičiaus ir projektų populiarumo, gali reikti pakankamai galingo serverio. Antrasis minusas, reikia patikimo serverio, nes esant serverio nepasiekiamumui, reklamos nebus rodomos, o tai neigiamai atsilies reklamos pardavimams, nes reklama turi pasiekti vartotojus.

Nagrinėjama reklamos valdymo sistema, skiriasi tuo, kad reklamos padalinamos serveriams, kurie jas rodo, kiekvienas serveris tampa pats atsakingas už reklamos parodymą, statistikos surinkimą. Šio modelio privalumai būtų tokie, kad reklamos rodomos visada, nepriklausomai nuo to ar veikia kiti serveriai, valdymo serveris nėra stipriai apkraunamas, centralizuotą reklamos valdymo sistemą galima netgi prijungti prie mažiau apkrauto projekto, taip sutaupant lėšų.

Pasirinkta reklamos valdymo architektūra sukelia vieną rimtą problemą, kaip realizuoti reklamų padalinimą serveriams ir kaip surinkti informaciją. Serverių pasiekiamumas gali būti nevienodas, mažai aprauti reklamos publikavimo serveriai, atsakymus gali paduoti greitai, daugiau apkrautas reklamos publikavimo serveris, gali ilgiau generuoti atsakymą. Nuosekliai patikrinti visus serverius yra paprasčiausias būdas, tačiau galima panaudoti ir lygiagrečius programavimo metodus, kad kreipiniai į serverius vyktų vienu metu. Ar šie metodai suteikia pranašumų, bus bandoma iširti šiame darbe.

Kriterijai, kuriais remiantis bus aptarinėjami metodai:

- Laikas – poveikis reklamų padalinimo laikui, statistikos surinkimo laikui;
- Metodo sudėtingumas – galimybė pritaikyti funkcijas kitiems skaičiavimams, papildomų serverio nustatymų poreikis, modifikavimo galimybės;
- Patikimumas – ar visuomet funkcija atlieka tai ką turi atlikti, ar pasitaiko nesklandumų;
- Keliamų reikalavimų kiekis – keliami reikalavimai kitai programinei ir techninei įrangai;

Centralizuota reklamos valdymo sistema, perkelta į realią aplinką, t.y. į internetinių projektų talpinimo paslaugas siūlančios kompanijos serverius. Kadangi tai yra virtualūs serveriai, tai gauti tikslesnę informaciją (procesoriaus apkrova, duomenų srautas) neįmanoma. Detalesnė tyrimo aplinka aprašyta 6 skyriuje.

Bandant įvertinti lygiagreto programavimo privalumus ir trūkumus, buvo pasirinkti 4 būdai perduoti informacijai. Kad būtų galima juos palyginti, visi jie buvo integruoti į sukurtą centralizuotą reklamos valdymo sistemą. Visi pasirinkti būdai detalai aprašyti 5 skyriuje.

2 Reklamos valdymo sistemų ypatumai ir naudojimas

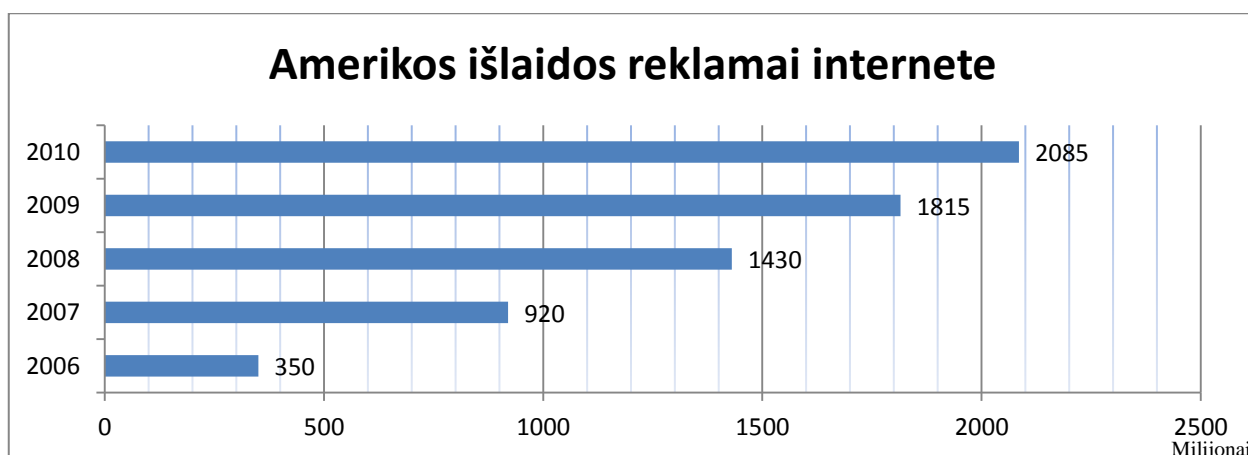
2.1 Reklamos valdymo sistemų svarba

Reklamos užsakovas, kuris suinteresuotas publikuoti reklamą kitose svetainėse, turi tris paprastus variantus. Reklamos užsakovas gali:

- Sutikti rodyti kito puslapio reklamas, mainais kad jo reklama bus rodoma pas juos;
- Mokėti reklamas transliuojantiems puslapiams;
- Mokėti kompanijoms, dažniausiai reklamų tinklams kaip DoubleClick ar Flycast, kad jie publikuotų reklamą pasirinktame svetainių skaičiuje. [11]

Reklamos užsakovai turi šiuos tris variantus, reklamas rodančių svetainių savininkas jų turi mažiau, kad užtikrinti pirmus du variantus užsakovams, reikia patogios reklamų valdymo sistemos, trečiuoju atveju ji nėra būtina tik todėl, kad visu valdymu rūpinasi trečiosios šalies programinė įranga.

Šiuo metu pasaulyje reklamai išleidžiama vis didesnės sumos pinigų, pagal www.emarketer.com surinktus duomenis pateikiama informacija apie reklamai išleistus pinigus Amerikoje. Pagal tas tendencijas, mes galima daryti išvadas, kad ir kitose pasaulio šalyse, rinkos pokyčiai yra panašūs.



1 Pav. Amerikos reklamos išlaidos

Tai parodo, kad reklama internete tampa vis aktualesnė ir jos dėka pasiekiamas vis didesnis vartotojų kiekis. Tam, kad užtikrinti patogų reklamos valdymą, naudojamos reklamos valdymo sistemos.

Jei jūs nenaudosite tinklo, arba mainų programos, jūs turėsite įdiegti technologiją, kad sekti vartotojų srautą, kad galėtumėte apmokestinti reklamos užsakovus teisingai. Daugėjant užsakovų, puslapių valdymas, srauto sekimas, pinigų surinkimas iš užsakovų, pareikalaus vis daugiau laiko ir pastangų, bet jeigu jūs pasiryžę plėsti savo puslapius, ir rodyti tik reklamas, kurios aktualios vartotojams, pelnas gali būti vertas pastangų. [14]

Pabandykite įsivaizduoti, kad valdote vieną puslapį ir jame yra trys pozicijos reklamai, reklama keičiasi kartą kas kelis mėnesius. Reklamos valdymas visai nesudėtingas, net ir turint paprasčiausią turinio valdymo sistemą. O kas pasidaro, kai turime penkis puslapius, kuriuose yra po dešimt ir daugiau reklaminių pozicijų, turime daugiau kaip dešimt užsakovų, kurių reklamos planai skirtingi? Suklydus rizikuojate netekti užsakovo ir pajamų, tad patogus, greitas, paprastas reklamų valdymas yra būtinas.

Pagrindiniai kriterijai, kuriuos turi atitikti reklamos valdymo sistema:

- Patogumas – reklamos publikavimas turi būti paprastas, administravimas lengvai randamas;
- Patikimumas – reklamos turi būti transliuojamos be nenumatytų trukdžių;
- Funkcionalumas – sistema kiek galima turi likti paprasta, bet tuo pačiu užtikrinti visas būtinas valdymo funkcijas.

Reklamos valdymo sistemų reikalingumą, galima palyginti su turinio valdymo sistemos svarba. Šiais laikais net ir maži projektai jungiami kartu su turinio valdymo sistema, nes tai suteikia didžiulį valdymo pranašumą.

Naršykle pagrįsti įrankiai, kuriais būtų galima kurti, keisti, trinti svetainės puslapius, ar atskirus skyrius [6].

Vis daugiau įrankių galima naudotis naršyklės pagalba, tai suteikia vartotojui daug didesnes galimybes ir laisvumą, nebėra pririšimo prie konkrečios techninės įrangos.

Svetainės pakeitimai gali būti atliekami akimirksniu, net nežinant HTML ar kitų programavimo kalbų. [6]

Vartotojams, nebėra keliami ypatingi reikalavimai, nereikia žinoti programavimo kalbos.

Šie požymiai buvo pritaikyti turinio valdymo sistemoms, tačiau turinio valdymo sistema nuo reklamos valdymo nelabai skiriasi, tad juos taipogi galima taikyti ir šiuo atveju.

2.2 Reklamų tipai ir reklamos veikimas

Dabartinė internetinė reklama, priešingai nei senosios paspaudžiamos reklamos, dažnai siūlo tas pačias pagrindines galimybes, tokias kaip paveikslukas, arba produkto pavadinimas, kompanijos pavadinimas, tekstas ir antraštė sudėta viskas į vientisą reklamą – internetinę reklamą.[3]

Remiantis šiuo internetinės reklamos apibūdinimu, galima daryti išvadą, kad reklama dabar yra daug interaktyvesnė, naudojamos naujos technologijos, siekiant sudominti vartotojus, daugumai šiuolaikinių reklamų naudojamos flash technologijos.

Kuriant reklamos valdymo sistemą būtina reikia atsižvelgti į šiuos reklamų tipus:

- Tekstinės reklamos
- Vaizdinės reklamos
- Vaizdo ir garso reklamos

Valdant reklamas, būtina turi būti galimybė patalpinti šio tipo reklamas, nes tai suteikia lankstumo kontroliuojant reklamines kampanijas.

2.3 Reklamos valdymo sistemų rinka ir vartotojai

Reklamos valdymo sistemas galima suskirstyti į mokamas ir atvirojo kodo. Dauguma reklamos valdymo sistemų veikia jau minėtu principu t.y. naudojamas reklamos serveris.

Vieni iš populiariausių nemokamų produktų yra „Google DoubleClick“ ir „OpenX“. Abu šie variantai turi ir privalumų, ir trūkumų. Abu produktai veikia reklamos serverių principu, tik pirmuoju atveju jums tuo neteks rūpintis, nes bus naudojami „Google“ serveriai.

Tai reklamos valdymo produktai, siūlomi visame pasaulyje bei plačiai paplitę Lietuvoje, nes jie turi geresnį palaikymą ir apie juos pateikiama daugiau informacijos.

Situacija Lietuvoje nėra aiški, informacijos yra nedaug. Pagrindiniai reklamos valdymo keliai yra du:

- Reklamos valdymas per TVS. Šias paslaugas siūlo beveik visos internetinius puslapius kuriančios kompanijos, tačiau reklamos valdymas yra labai paprastas ir nefunkcionalus, jis skirtas tik konkretaus puslapio informacijai pakeisti.
- Reklamos apsikeitimo svetainės. Tai Lietuvoje populiarėjantis būdas, nes nereikalauja ieškoti reklamos užsakovų. Tokias paslaugas teikia www.skydeliai.lt, www.flylinks.lt, ir t.t.

Lietuvoje reklamos internete rinka vis labiau auga, svetainės aktyviai talpina pas save įvairaus tipo reklamas. Reklamos valdymo sistemos vartotojus galima būtų suskirstyti į:

- Reklamos rinkos lyderiai. Tai vartotojai, kurie turi daug reklamos užsakymų ir daug publikuojančių svetainių. Reklama yra pagrindinis jų pelnas. Dažniausiai jie naudoja individualiai kurtas reklamų valdymo sistemas, kurios yra pritaikytos jų poreikiams.
- Reklamos rinkos vidutiniokai. Tai vartotojai, kurie turi 3-10 svetainių, reklamos nėra labai daug. Jiems investuoti į nuosavą valdymo sistemą dar nėra tikslo.
- Reklamos rinkos naujokai. Turi keletą svetainių ir nedaug reklamos. Jiems pilnai užtenka atvirojo kodo sistemų, dažniausiai „Google“, nes neturint savos reklamos, tuščias vietas galima užpildyti „Google“ reklamomis.

Kurta programinė įranga buvo orientuota į vidutinę vartotojų kategoriją, nes su centralizuota reklamos valdymo sistema jiems galima garantuoti nesunkų valdymą, reikiamą funkcionalumą ir stabilumą.

3 Lygiagretaus programavimo metodų panaudojimo galimybės valdant reklamos publikavimo serverius

Pirmiausiai norėčiau paminėti kodėl visgi PHP kalba.

Šiuo metu PHP programiniai kodai naudojami milijonuose interneto svetainių, o joms kurti naudojama atvirojo kodo politika leidžia bendromis daugelio programuotojų jėgomis nuolat ją tobulinti. [5]

PHP yra nuolat atnaujinama ir šiuo metu plačiai naudojama 5 versija. Šiuo atveju pasirinkta PHP kalba, nes:

- PHP programos gali būti vykdomos įvairių OS (Windows, UNIX, Linux) bazėje dirbančiuose serveriuose.[5]. Taigi mums nekyla problemų dėl suderinamumo, sistema puikiai veiks bet kuriame serveryje, kuriame yra PHP5.
- PHP procesoriai ir kitos pagalbinės priemonės parengti atviro kodo principu ir laisvai platinami internete.[5]. Priemonės programos kūrimui nieko nekainuoja, tai viena iš priežasčių, kodėl PHP yra tokia populiari.
- Pritaikyta informaciniams ryšiams su populiaria ir laisvai platinama duomenų bazių valdymo sistema MySQL. [5]. Kadangi duomenims saugoti pasirinkta būtent ši duomenų bazių valdymo sistema, tai yra patogiu suderinti šias dvi priemones.
- Dabar tai galinga objektinio programavimo kalba, kuria galima aprašyti sudėtingus interneto technologijų procesus. [5]

Taigi PHP kalba mums suteikia visas galimybes, kurių mums reikia. Tai ir atvirojo kodo programinė įranga ir didelis pasirinkimas veikiančių serverių, kurie gali užtikrinti reikiamą techninę ir programinę bazę tyrimui.

PHP programavimo kalba teoriškai nesuteikia galimybių lygiagrečiam programavimui, nes nepalaiko gijų. Tačiau pasinaudojant tam tikromis funkcijomis, galima pasiekti, kad sistema veiktų ir atliktų veiksmus lygiagrečiai.

Pasiekti tai, kad veiksmai būtų pradėti vykdyti vienu metu, galima pasinaudojus „Cron Jobs“, arba AJAX. Abiem atvejais tai galima padaryti, bet detaliau tai nebus analizuojama, nes tai ganėtinai specifiniai būdai, kurie ne visais atvejais gali būti pritaikomi. „Cron jobs“ yra labiau skirtas užduotims atlikti numatytu laiku. O AJAX yra skirta informacijai atnaujinti svetainėse, ar jų dalyse neperkraunant visos svetainės turinio.

Man buvo aktualiau išbandyti PHP kalbos galimybes šioje srityje ir pabandyti pasiekti, kad sistema atliktų veiksmus lygiagrečiai, naudojantis šios programavimo kalbos funkcijomis.

Pasirinkti būdai yra detaliam aprašyti 5 skyriuje. Buvo bandomi du lygiagretūs metodai ir du nuoseklūs, taip buvo galima tiksliai juos palyginti ir objektyviai įvertinti, teik panaudojimo sudėtingumą, tiek taikymo galimybes, tiek ir jų veikimo greitį.

3.1 Lygiagretaus programavimo metodų naudojimo plusai

Pagrindinis, kriterijus, kuo naudingas yra lygiagretus programavimas, yra laikas. Reklamos valdymo sistemos atveju, tiek siunčiant informaciją, tiek ją gaunant, sujungiama su keliais reklamos talpinimo serveriais. Vienam iš jų esant labiau apkrautam ir nenaudojant lygiagretaus programavimo, visi už jo esantys serveriai turės laukti savo eilės, kol informacija bus gauta, arba perduota. Naudojant lygiagretaus programavimo galimybes, į visus serverius užklausa išsiunčiama vienu metu ir vartotojo pradėta funkcija užtrunka panašiai tiek, kiek užtrunka ilgiausiai rezultatų negražinęs, ar nepasiėmęs serveris.

3.2 Lygiagretaus programavimo metodų naudojimo minusai

Lygiagretus programavimas yra sudėtingas, pasiūlyti jį pradėdantiesiems, tai tas pats kas prašymas Apache serverių užlūžimo. [7]

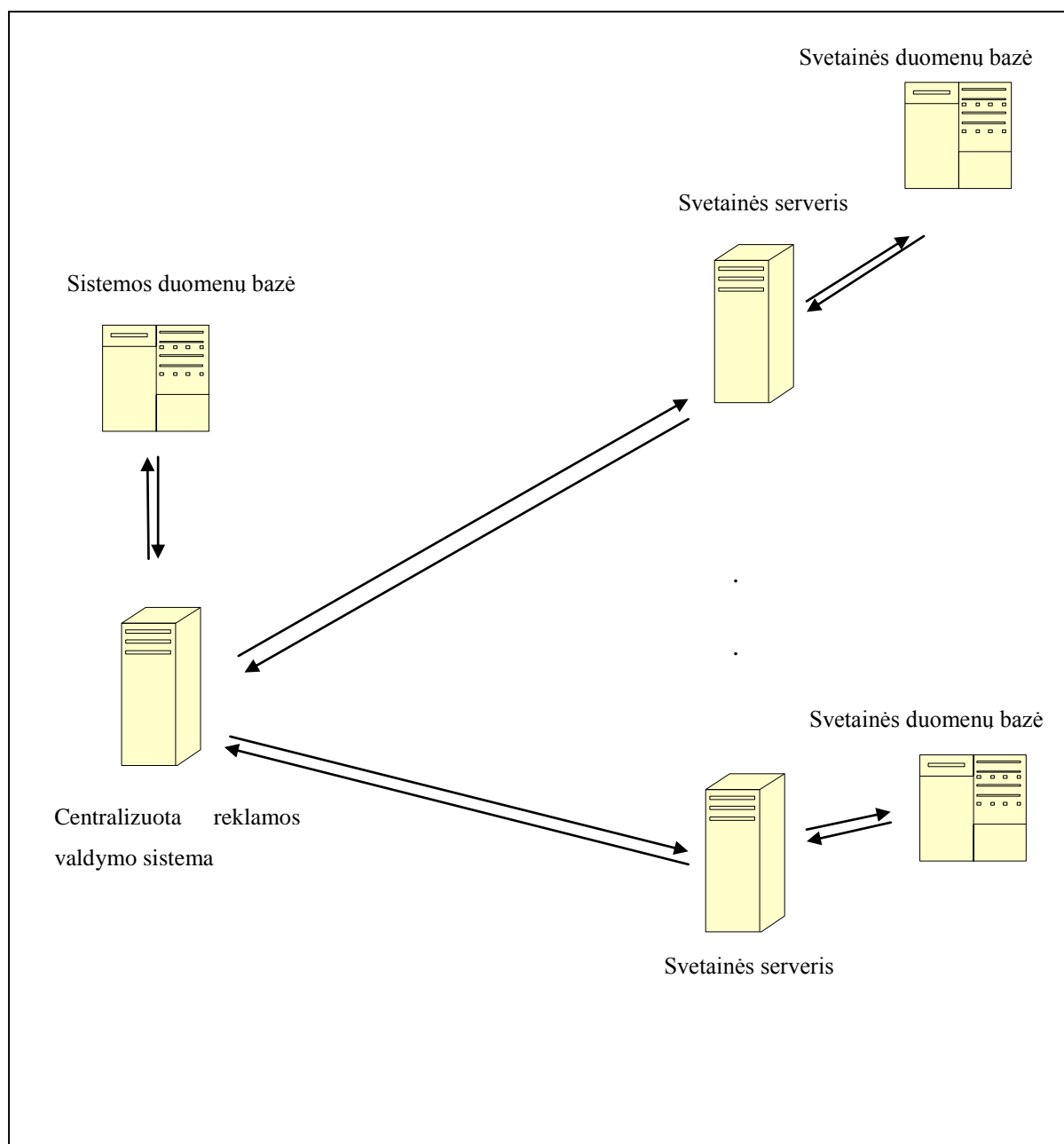
Pasiekti norimą funkcionalumą naudojant eilines PHP kalbos funkcijas yra žymiai lengviau, nei pasiekti tą patį funkcionalumą, naudojantis lygiagretaus programavimo galimybėmis.

Programas kurtas naudojant lygiagretų programavimą, yra labai sunku išbandyti, surasti klaidas ir palaikyti.[4].

Testavimas tampa sudėtingesnis, sunkiau nuspėjamas sistemos veikimas. PHP kalboje lygiagretus programavimas nėra labai plačiai paplitęs, tad ne daugelis su juo susiduria. Jeigu programoje yra panaudota daug specifinių funkcijų, kitam programuotojui, ar net ir kūrėjui, gali būti sudėtingiau įvesti naujus patobulinimus, ar ištaisyti klaidas.

4 Centralizuotos reklamos valdymo sistemos aprašymas

2 paveiklėlyje yra pabvaizduotas centralizuotos reklamos sistemos architektūrinis išdėstymas ir bendras veikimo principas.



2 Pav. Centralizuotos reklamų valdymo sistemos architektūra

Centralizuotos reklamos valdymo sistemos veikimo principas:

Sistema įdiegiama pasirinktame kliento serveryje. Sistema nėra absoliučiai nepriklausoma, projektas prieš prijungiant turi būti pritaikytas, turi būti numatytos reklamos pozicijos, bei sukurta duomenų bazė. Prisijungus prie sistemos bus galima prijungti norimus portalus. Informacija apie esančias reklamas portaluose bus surenkama jungiantis prie kiekvieno serverio atskirai ir patikrinant įrašus duomenų bazėse. Radus reklamų, jų informacija atkeliaus

į sistemos duomenų bazę. Informacijos surinkimas nebus vykdomas kiekvieną kartą, o tik vartotojui pasirinkus, nes jeigu bus dirbama tik su šia reklamos valdymo sistema, tai paskutinio atlikto pakeitimo duomenys bus likę sistemoje ir darbas bus tęsiamas toliau. Reklaminė informacija bus pateikiama vartotojui. Atlikus pakeitimus, visi nauji pasikeitimai bus atnaujinti svetainių serveriuose, atnaujinant jų duomenų bazes ir persiunčiant reklamas.

Pasirinkta architektūra nėra būdinga reklamos valdymo sistemoms, nes ji sukelia šias problemas:

- Sudėtingas statistikos valdymas
- Sudėtingas reklamų valdymas ir jų rodymas svetainėse
- Užsakovų ir statistikos apjungimas

Tam, kad išspręsti šias problemas ir pagreitinti bendravimą su nutolusiais serveriais (reklamas publikuojančiomis svetainėmis), nuspręsta naudoti lygiagretaus programavimo metodus.

Taip buvo pagreitintas statistikos surinkimas, tuo pačiu statistikos apdorojimas perduotas reklamas rodančiai svetainei, reklamos valdymo sistemai paliekant tik statistikos apjungimą.

Kadangi pagreitėja surinkimas, atsiranda realesnė galimybė statistiką surinkinėti dažniau.

Reklamų valdymas vyksta sistemoje ir tik aktyvių reklamų kopijos išsiuntinėjamos po svetainės norimu metu. Naudojant lygiagretų programavimą bandoma sutrumpinti išsiuntimo laiką.

5 Centralizuotos reklamos valdymo sistemos funkcijos ir galimybė pritaikyti lygiagretų programavimą

Valdant sukurtą centralizuotą reklamų valdymo sistemą, nėra jaučiama pasirinktos architektūros įtaka, nes visa informacija yra ne tik padalinta skirtingiems serveriams, bet ir sukaupta sistemos serveryje. Didžiausia pasirinktos architektūros įtaka jaučiama norint atnaujinti reklamas ir surinkti statistiką rankiniu būdu.

Pagrindinės centralizuotos reklamos valdymo sistemos funkcijos:

- **Reklaminės informacijos surinkimas** – pasinaudojant lygiagretaus programavimo principais, jungiamasi prie vartotojo turimų svetainių ir surenkama ten turima reklamų informacija, šiuo atveju parsiuočiama visa informacija apie reklamas, jų parametrai ir patys paveikslukai. Šis veiksmas nėra nuolat atliekamas vartoto prisijungimo metu, o tik vartotojui laisvai pasirinkus. Jeigu pastebima, kad pasikeitė kurio nors vartotojo puslapio reklaminė informacija, vartotojo reikalaujama atsinaujinti reklamas, prieš tęsiant tolimesnius darbus;
- **Reklaminės informacijos atnaujinimas** - pasinaudojant lygiagretaus programavimo principais, jungiamasi prie serverių, kuriems buvo atlikti pakeitimai ir atnaujinama informacija juose, perkeliama reklaminiai paveikslukai, ar kita reikalinga informacija. Reklaminės informacijos atnaujinimas nėra būtinas, vartotojo atlikti keitimai yra saugomi nuolat, tad kitą kartą prisijungęs, jis mato savo paskutinio darbo rezultatus.
- **Svetainės užregistravimas** – vartotojas gali turėti neribotą skaičių svetainių, jeigu tai nėra apribota administratoriaus. Svetainės registravimo metu surenkama visa aktuali informacija. Šie duomenys yra saugomi užkoduoti MySQL duomenų bazėje.
- **Svetainės redagavimas** – pakeičiama svetainės informacija.
- **Svetainės šalinimas** – svetainės pašalinimas yra galimas tik tuo atveju, jeigu visos reklaminės vietos, susietos su šia svetaine yra pašalintos, taip yra tam atvejui, kad nebūtų pašalinta ne ta svetainė.
- **Reklaminės vietos kūrimas** – vartotojas pasirenka, kokio dydžio reklama bus toje vietoje, kokie tos vietos specialūs parametrai, t.y. reklamos tipų draudimai. Informacija išsaugoma ir parodoma detali informacija, kas turi būti padaryta pačiame portale, t.y. kokio tipo nuoroda turi būti patalpinta svetainėje, kad ten būtų rodomos šios vietos reklamos.
- **Reklaminės vietos šalinimas** – pašalinama reklaminė vieta, kartu su joje esančiomis reklamomis.

- **Reklaminės vietos redagavimas** – pasirenkamas kitas reklaminės vietos dydis, bei specialūs parametrai.
- **Ataskaitų generavimas** – vartotojas gali sugeneruoti bet kurios svetainės detalią išklotinę, ataskaita atidaroma atskirame lange peržiūrai, ji yra suformuota spausdinimui.
- **Reklamos įdėjimas** – pasirenkama svetainė, pasirenkama reklamos vieta, nustatomi reklamos nustatymai t.y. kiek kartų ją parodyti, arba iki kada rodyti, prioritetas (jeigu toje vietoje yra daugiau reklamų) ir t.t. Pasirenkamas reklamos failas (SWF,GIF,JPEG,...), arba JavaScript kodas, kad būtų galimybė krauti reklamą iš kitų vietų. Jeigu dedamas paveikslukas, jis sumažinamas (jeigu per didelis) iki reklaminės vietos dydžio.
- **Reklamos redagavimas** – galima keisti visą informaciją, kuri yra suvedimo metu.
- **Reklamos šalinimas** – reklama ištrinama.
- **Reklamos blokavimas** – reklama nerodoma kol nėra atblokuota, yra galimybė nustatyti laiką, kada jos blokavimas baigiasi.
- **Vartotojo įkėlimas** – tik administratorius gali įkelti naują vartotoją, administratorius perduoda prisijungimas portalo savininkui. Įkėlimo metu, vartotojui nustatomi apribojimai, sukuriama prisijungimo vardas ir paskiriamas slaptažodis.
- **Vartotojo redagavimas** – vartotojo informacijos pakeitimas, galima pakeisti vartotojo slaptažodį, net ir tuo atveju, jeigu nėra žinomas to vartotojo slaptažodis.
- **Vartotojo blokavimas** – Vartotojas užblokuojamas, informacija apie jį išlieka sistemoje, tačiau tas vartotojas nebegali prisijungti.
- **Vartotojo šalinimas** – Pašalinama vartotojo informacija iš sistemos.

5.1 Centralizuotos reklamos valdymo sistemos funkcijos, kurios veikia naudojant lygiagrečius metodus

Funkcijos, kurios veikia naudojant lygiagrečius programavimo metodus:

- Reklamų išsiuntimas
- Svetainių patikrinimas
- Statistikos parsisiuntimas

Visais trim atvejais sistema jungiasi prie išorinių serverių, kad išsiųsti, ar gauti atitinkamą informaciją. Panaudojus lygiagrečius metodus, sumažinama įtaka tų serverių, kurie yra labai apkrauti, ir kurie vėluoja gražinti ar gauti rezultatus.

Veikimo principas panašus visoms funkcijoms. Veiksmai vykdomi šia seka:

1. Išsiunčiamas užklauskimas serveriui, kad gražintų patvirtinimo kodą;
2. Išsiunčiama atitinkama komanda, priklausomai nuo funkcijos;
3. Svetainė paruošia rezultatus;
4. Nuskaitomi gražinti iš serverio duomenys;
5. Nutraukiamas ryšys su svetaine;

5.2 Centralizuotos reklamos valdymo sistemos funkcijos, kurias galima pritaikyti naudoti lygiagrečius metodus

Sistemoje esančios funkcijos, kurioms dar galim pritaikyti lygiagretų programavimą:

- Užsakovų informavimo funkcija. Elektroniniai laiškai gali būti išsiunčiami greičiau, ypatingai tai turėtų būti jaučiama, jeigu siuntimas būtų vykdomas nustatytu laiku, ir kiekvienam užsakovui laiškas būtų generuojamas personaliai.
- Statistikos apdorojimas. Šiuo metu sistema renka tik labai nedidelį statistikos kiekį, ateityje bus renkama žymiai detalesnė informacija, todėl ir jos apdorojimas užims daugiau laiko.

Panaudoti lygiagrečius metodus galima plačiau, sistemai plečiantis gali atsirasti ir daugiau sudėtingų algoritmų, kurie gali pareikalauti daugiau serverio resursų. Norint to išvengti dalį užduočių būtų galima lygiagrečiai paskirstyti nutolusiems sistemos serveriams, kurie atliktų skaičiavimus ir gražintų rezultatus.

6 Lygiagretaus programavimo metodai, naudojami analizuojant reklamos valdymo sistemą

Žmonės dažnai daro prielaidą, kad norint atlikti kelis darbus vienu metu, reikia naudoti gijas – ir tada jie supranta, kad PHP nepalaiko gijų, tuomet jie pereina prie kažko kito, kad ir PERL. [10]

Bet kelių darbų atlikimą vienu metu su PHP pasiekti galima ir netgi keliais variantais. Aptarsiu kelis variantus, kuriais galima pasiekti norimo rezultato ir palyginimui pateiksiu dar kelis variantus, kuriais tas pats rezultatas bandomas gauti nuosekliai.

6.1 CURL biblioteka

Neskaitant to, kad cURL naudojama gauti informacijai iš kitų svetainių, yra galimybė naudoti cURL lygiagrečiam darbui su PHP pasiekti. [12]

cURL biblioteka sukurta tam, kad būtų galima patogiau ir paprasčiau bendrauti su kitais serveriais. Sukurti bendravimo protokolai leidžia gauti informaciją iš įvairių tipų serverių, bet mums aktualiausias tik HTTP protokolas.

Naudojamos funkcijos:

- `curl_multi_init`
- `curl_multi_add_handle`
- `curl_multi_exec`
- `curl_multi_getcontent`
- `curl_multi_remove_handle`
- `curl_multi_close`

Funkcijos naudojamos sukurti ryšiui su reikiamomis svetainėmis. Pasinaudojant funkcija `curl_multi_exec` išsiunčiama užklausa į norimas svetaines. Po išsiuntimo nelaukiama, kol bus gautas rezultatas, o tęsiamas funkcijos vykdymas, tai dirbtinai sukuriama daug gijų, iš kurių informacija gaunama priklausomai nuo to, kuri pirmiau ją gražina. Informacija nuskaitoma su `curl_multi_getcontent` funkcija ir išsaugoma rezultatų masyve. Sujungimai su svetainėmis nutraukiami, kai tik visa informacija iš reikiamos svetainės pasiekia sistemą.

cURL privalumas lyginant su kitais variantais, tai jos naudojimo paprastumas, nes tai yra atskira biblioteka, kur jau ilgą laiką standartiškai diegiama į daugumą serverių ir plačiai naudojama.

Vienintelis reikalavimas, kad būtų galima naudoti šią biblioteką, tai, kad ji būtų įdiegta serveryje ir aktyvi.

6.2 Stream_select funkcijos panaudojimas

stream_select – veikia panašiai, kaip select() funkcija. Sistema kviečia pagal pateiktą sujungimų masyvą su nustatytu veikimo laiku, nurodytu tv_sec ir tv_usec parametrais. [8]

Stream_select veikimas yra labai paprastas, sukūrus prisijungimus, laukiama rezultatų. Kai tik atsiranda, ką skaityti, atiduodama skaitymui. Veikimas teoriškai kažkuo panašus į nuoseklaus veikimo funkcijas, tačiau skirtumai yra šie:

- Veikia lygiagrečiai, laukia informacijos iš bet kurio šaltinio, nuskaito iš tų, kurie gražina informaciją
- Nenaudoja procesoriaus resursų, kol laukia.

Antrasis teiginys apie funkciją, yra labai svarbus, nes procesoriaus apkrovimas gali sukelti kitų projektų sutrikimus.

Atkreipkite dėmesį, kad laukimas neturi įtakos serverio procesoriui. Nėra nieko keisto, kai tinkle veikiančios programos, kurios naudoja fread funkciją cikle, dažnai procesoriaus apkrovimą padidina iki 100 procentų. Tačiau tai nėra stresam_select funkcijos problema, ji turi reikiamus nustatymus, kurių dėka reaguoja nedelsiant, kai yra ką skaityti ir neapkrauna procesoriaus laukimo metu. [1]

Ši stresam_select savybė yra labai svarbi ir aktuali tiems, kas nenori investuoti daug į techninę įrangą, o resursai yra riboti.

Pagrindinės naudojamos funkcijos:

- Stresam_socket_client
- fwrite
- Stresam_select
- fread
- fclose

Galim pastebėti, kad tiek rezultatų skaitymui, tiek parametrų perdavimui ir ryšio nutraukimui, naudojamos paprastos PHP kalbos funkcijos, kurias taip pat sėkmingai galima panaudoti ir sujungiant svetaines nuosekliai. Pagrindinis skirtumas yra ryšio sukūrimas, kuris transliuoja rezultatus tol, kol jie nėra nuskaityti, nuskaitymas vykdomas blokais, todėl, net jeigu visa informacija nėra nuskaityta iš karto, ji papildoma vėliau. `Stream_select` funkcija užtikrina tai, kad nuskaitymas vyktų iš ten, kur greičiausiai paduodami rezultatai. Taigi visos svetainės yra sujungtos lygiagrečiai ir visos vienu metu gražina duomenis.

`Stream_select` naudojimui yra būtina PHP5 versija, šis reikalavimas yra nesunkiai įgyvendinamas, nes, tai nebėra naujiena ir dauguma komercinių serverių dažnai atnaujina savo programinę įrangą.

6.3 Fopen funkcijos panaudojimas

`fopen` – atidaro failą pagal URL adresą. [13]

Tai pats paprasčiausias būdas gauti informacijai. Funkcija kreipiasi į nurodytą adresą ir laukia rezultatų, kol nėra gavusi atsakymo, ji sustabdo tolimesnį algoritmo veikimą, todėl jeigu serveris neatsako ilgesnį laiką, algoritmo įvykdymas užtrunka. Kad nuskaityti rezultatus, naudojamos kitos PHP funkcijos, kad ir ta pati kaip ir jau minėta `fRead`, kuri nuskaitytu duomenis ir perduoda į „string“ tipo kintamąjį.

Vienas iš šios funkcijos privalumų, atidarant failą, galima nurodyti kas su juo bus daroma, t.y. atidaryti tik skaitymui, arba tik rašymui, arba abiem veiksmais.

Šiuo būdu sujungti svetainės yra paprasta, šiuos metodus galima plačiau pritaikyti ir kitų duomenų skaitymui ar rašymui, funkcijų naudojimui nėra keliami jokie papildomi reikalavimai, nes tai yra standartinės PHP kalbos funkcijos.

6.4 File_get_contents funkcijos panaudojimas

`file_get_contents` – nuskaityti visa failą į „string“ tipo kintamąjį [16]

Ši funkcija yra naudojama paprasčiausiai, nes nereikia rūpintis nei failo atidarymu, nei uždarymu, šiai funkcijai nurodžius URL adresą, rezultatai iškart išsaugomi į paskirtą kintamąjį ir juos galima apdoroti.

Minusas, kaip ir `fOpen` funkcijos atveju, kol rezultatai nėra išsaugoti, tol tolesnis algoritmo veikimas yra sustabdytas, taigi veiksmai atliekami nuosekliai.

Funkcijos privalumas, tai labai paprastas naudojimas ir galimybė pritaikyti nuskaitant informaciją iš įvairių šaltinių.

Tai yra standartinė PHP kalbos funkcija, todėl dėl jos naudojimo nėra keliami jokie reikalavimai.

7 Pasirinkta tyrimo aplinka

Tyrimui atlikti buvo naudojami virtualūs serveriai, kuriuose dar papildomai buvo sukurti „subdomenai“. Ši tyrimo aplinka buvo pasirinkta dėl šių priežasčių:

- Virtualūs serveriai neturi paskirstytų resursų. Tai reiškia, kad jų apkrova kinta ir dėl kitų projektų, esančių tame pačiame serveryje. Kadangi į reklamas rodančias svetaines nėra didelio lankytojų srauto, tai papildoma neprognozuojama serverio apkrova sukuria tikresnę serverio vėlinimą.
- Sukurti „subdomenai“ veikia kaip atskiri serveriai. Kadangi, kaip minėjau virtualiame serveryje yra daug projektų, tai ir „subdomenai“ dalinasi tais pačiais resursais, turi atskirus prisijungimus prie duomenų bazės ir mažai kuo skiriasi nuo atskirų virtualių serverių.

Taip buvo gauta pakankamai serverių testavimui ir tyrimui, kurie atitiko keliamus reikalavimus ir užtikrino reikiamą patikimumą už priimtina kainą.

Serverių informacija:

- PHP versija: 5.2.12
- MySQL versija: 5.0.89

Svarbiausi serverio nustatymai, kurie buvo reikalingi:

- Įdiegta cURL biblioteka
- PHP 5 versija
- Galimybė kurti „subdomenus“

Informacija apie naudojamą techninę įrangą nėra aktuali.

Programinės įrangos versijos ir nustatymai atitinka visus keliamus reikalavimus, kad visi naudojami metodai ir funkcijos veiktų korektiškai.

Tyrimui atlikti buvo iš viso sukurta 10 svetainių, kuriose priklausomai nuo atliekamo skaičiavimo buvo talpinama nuo 1 iki 15 reklamų.

Reklaminis paveikslukas visose svetainėse buvo naudojamas tas pats, t.y. PNG tipo paveikslėlis, kurio užimama vieta serveryje yra 26,9KB.

Tyrimo aplinka buvo sukurta taip, kad matuojant laiką, būtų matomas laiko pokytis, kuris priklausytų tik nuo svetainių, arba reklamų skaičiaus, bei nuo serverių apkrovimo. O reklaminės pozicijos, reklamos tipai ir reklamos užimama vieta serveryje būtų neaktuali.

8 Atliekamo tyrimo rezultatai ir rezultatų analizė

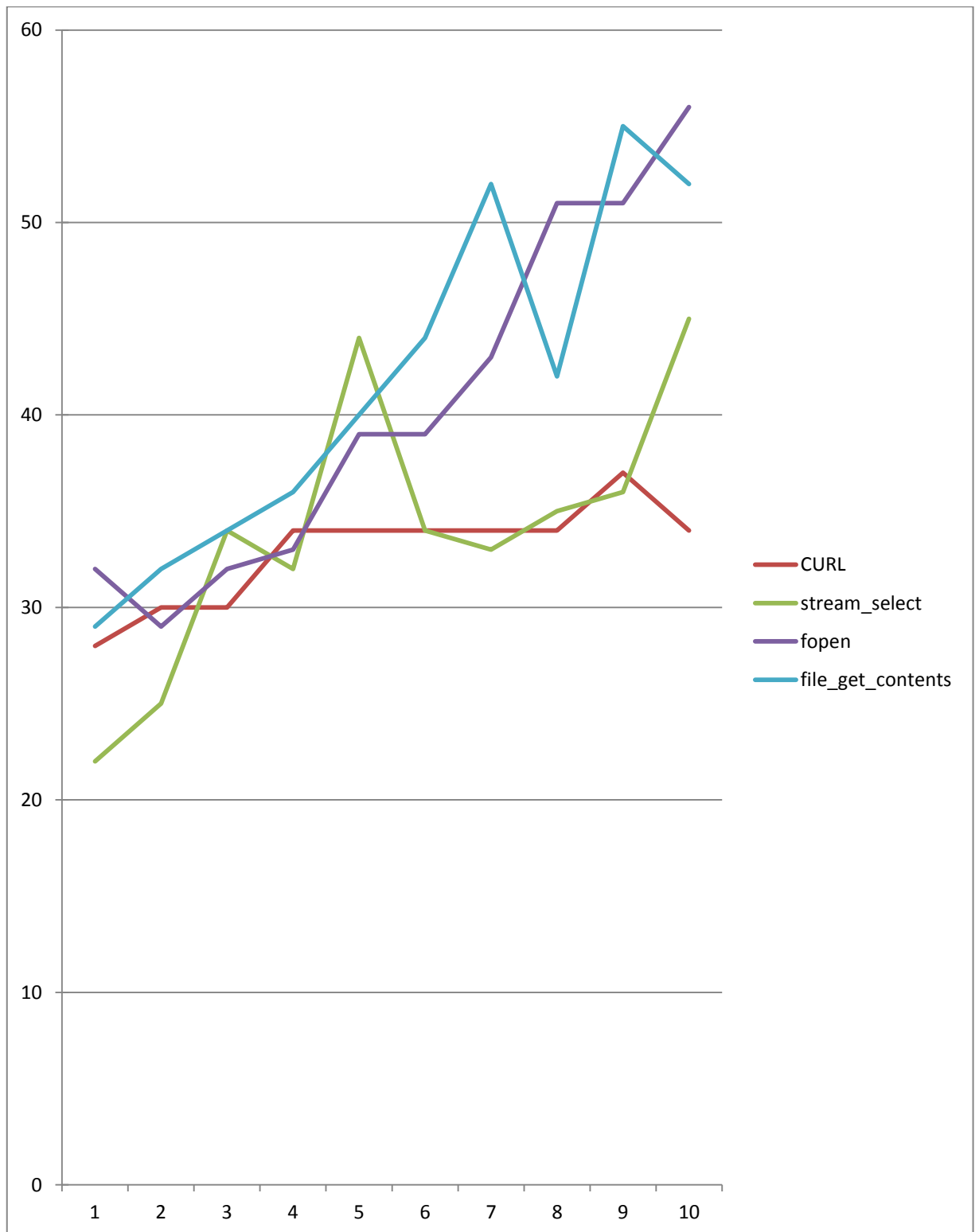
Detali informacija apie tyrimo metu atliktus laiko matavimus pateikiama prieduose. Kiekvienas matavimas buvo atliekamas tris kartus ir iš visų trijų išvedamas vidurkis, taip buvo bandoma sumažinti netikėtų laiko šuolių, ar bent jau juos truputį nuslopinti bendrame kontekste.

8.1 Laiko pokytis priklausomai nuo svetainių skaičiaus

Matuojamas laikas buvo, kai svetainėse yra po penkias reklamas, visi trys matavimai šiek tiek skirtingi.

Matuojant svetainių patikrinimo funkcijos greitį, svetainėse funkcija gražina tik patvirtinimo kodą, tad informacijos patikrinimo metu nėra kreipiamasi į svetainių duomenų bazę, ar perkeliama kokia nors informacija. Tai greičiausiai įvykdoma funkcija.

Detali informacija pateikta 2 lentelėje, o rezultatų atvaizdavimas grafiku 3 paveiksle.



3 Pav. Skirtingų metodų įtaka laikui. Svetainių patikrinimas.

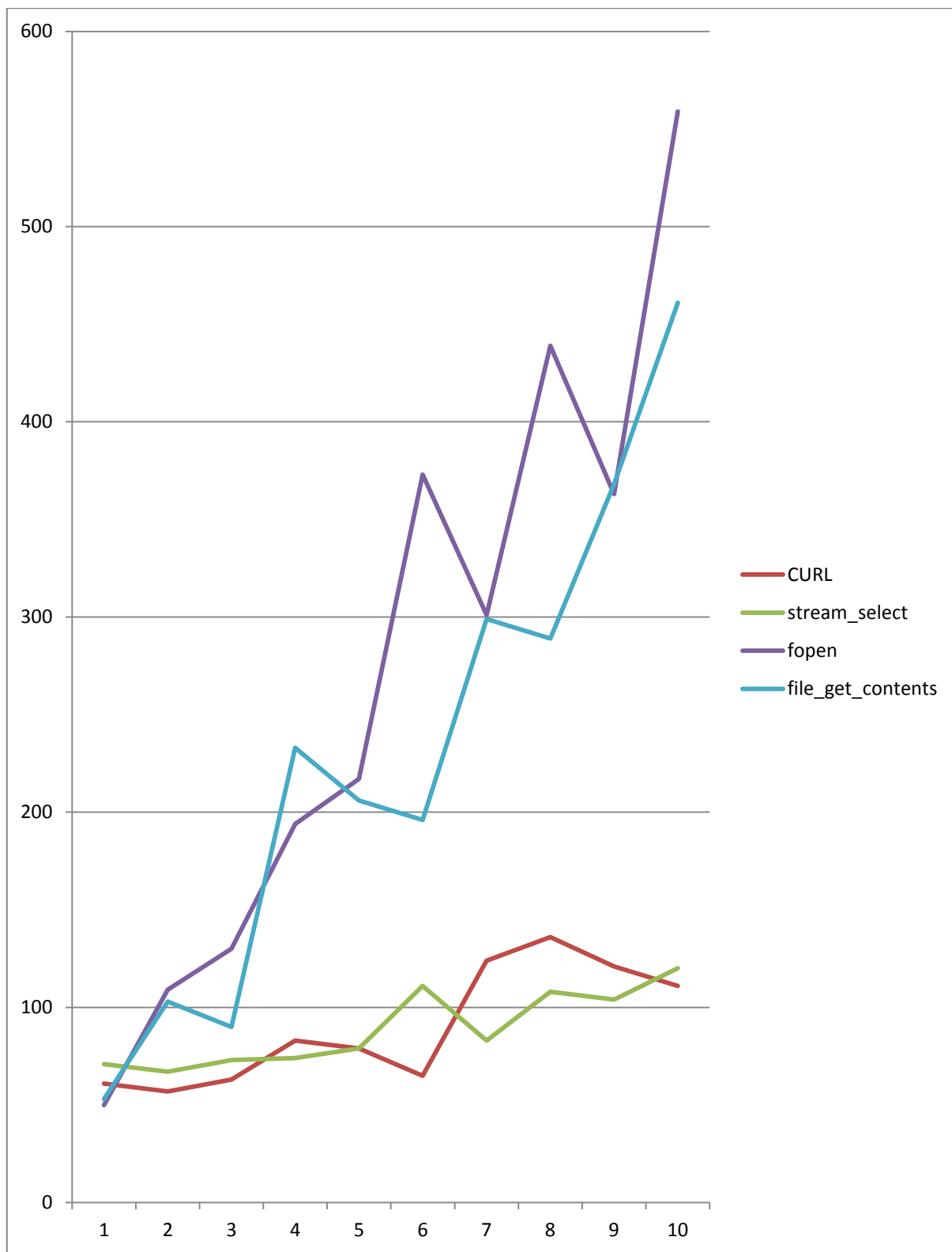
Prijungus 1 ar dvi svetaines, visų metodų rezultatai yra ganėtinai panašūs ir yra apie 30 milisekundžių. Jungiant didesnę kiekį svetainių pradeda ryškėti lygiagrečių metodų pranašumas, nes jie informaciją surenka daug greičiau.

Pajungus visas 10 svetainių laiko skirtumas tampa beveik dvigubu.

Grafike matosi, kad funkcija, naudojant lygiagrečius metodus, įvykdoma stabiliau, yra mažiau šokinėjimų ir įvykdymo laikas jungiant vis daugiau svetainių išlieka panašus, o naudojant nuoseklų būdą, kuo daugiau svetainių jungiama, tuo laikas auga.

Antrasis laiko matavimas yra labiausiai apkraunantis sistemą, ir užtrunkantis žymiai ilgesnį laiką, perkelti reklamas iš sistemos perduodama informacija, kuri įrašoma į svetainių duomenų bazę ir persiunčiamos reklamos.

Detali informacija pateikta 3 lentelėje, o rezultatų atvaizdavimas grafiku 4 paveiksle.



4 Pav. Skirtingų metodų įtaka laikui. Reklamų perkėlimas.

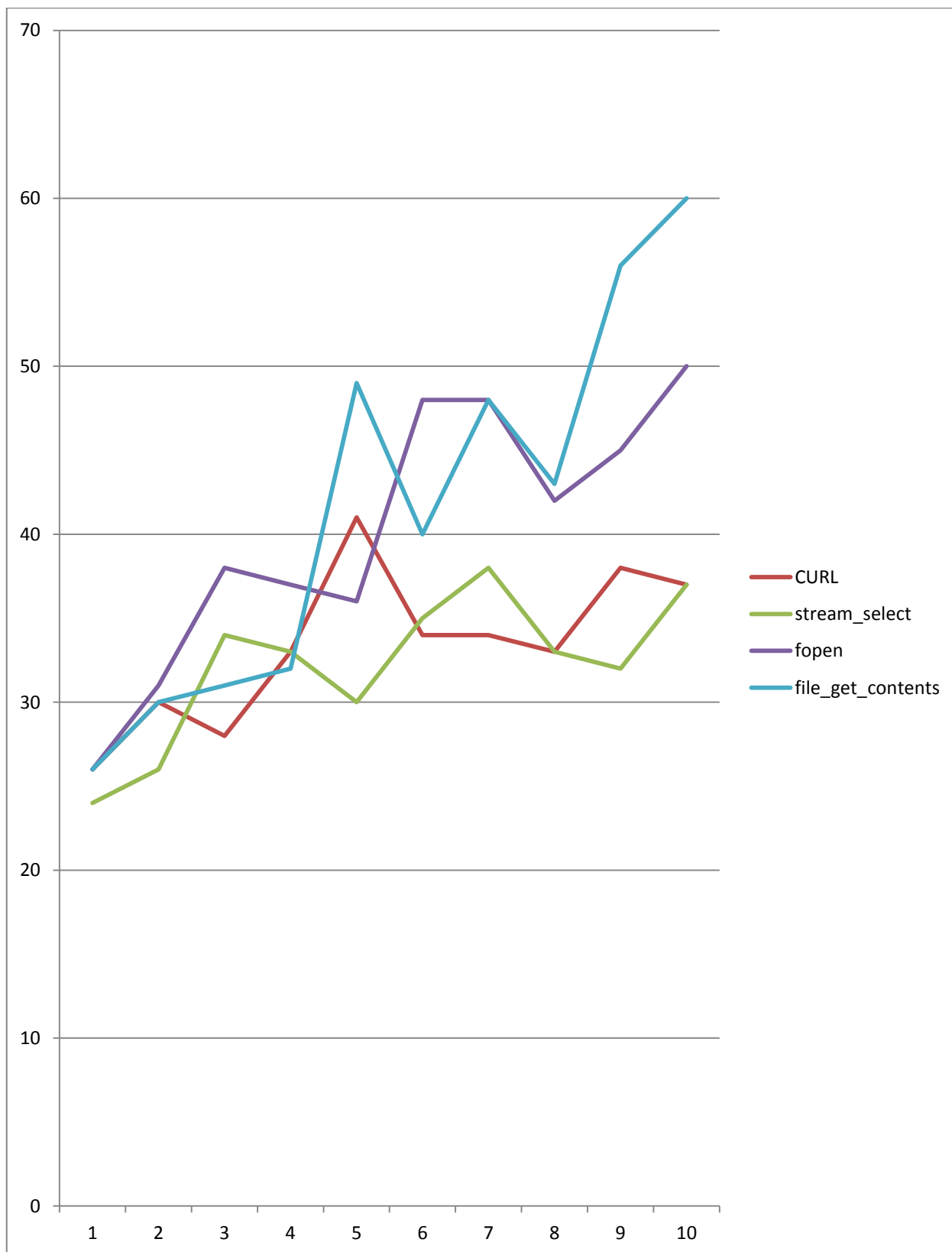
Prijungus vieną svetainę nuoseklus programavimas netgi turi pranašumą prieš lygiagretų: dėl paprastesnio algoritmo funkcija atliekama greičiau. Nuosekliems metodams reikia maždaug 50 milisekundžių laiko, o lygiagretiems 60-70 milisekundžių.

Jungiant daugiau svetainių pradeda ryškėti lygiagrečių metodų privalumai. Funkcijos atlikimo laikas žymiai stabilesnis, atlikimo laikas taipogi mažiau priklauso nuo prijungtų svetainių skaičiaus. Nuosekliai pajungus svetaines funkcija atliekama ilgai. Dešimt svetainių ir penkios reklamos negali būti laikomos sistemos riba, tad norint įkelti daugiau reklamų, atnaujinimo laikas gali labai stipriai pakilti. Kadangi serveriuose būna nustatytas vykdymo laikas, bandant perkelti per daug svetainių, ši riba nuosekliai pajungtose svetainėse bus pasiekta žymiai greičiau.

Pajungus 10 svetainių lygiagrečių metodų pagalba funkcija atliekama per maždaug 100 milisekundžių, kai tuo tarpu nuosekliai pajungtos svetainės užtrunka gerokai ilgiau. Įvykdymo laikas skiriasi penkis kartus. Jungiant dar daugiau svetainių šie skaičiai kistų dar greičiau.

Trečiasis atvejis yra labiau panašus į pirmąjį, nes gražinant statistiką, svetainėse tik kreipiamasi į duomenų bazę ir perduodami suformuoti duomenys.

Detali informacija pateikta 4 lentelėje, o rezultatų atvaizdavimas grafiku 5 paveiksle.



5 Pav. Skirtingų metodų įtaka laikui. Statistikos surinkimas

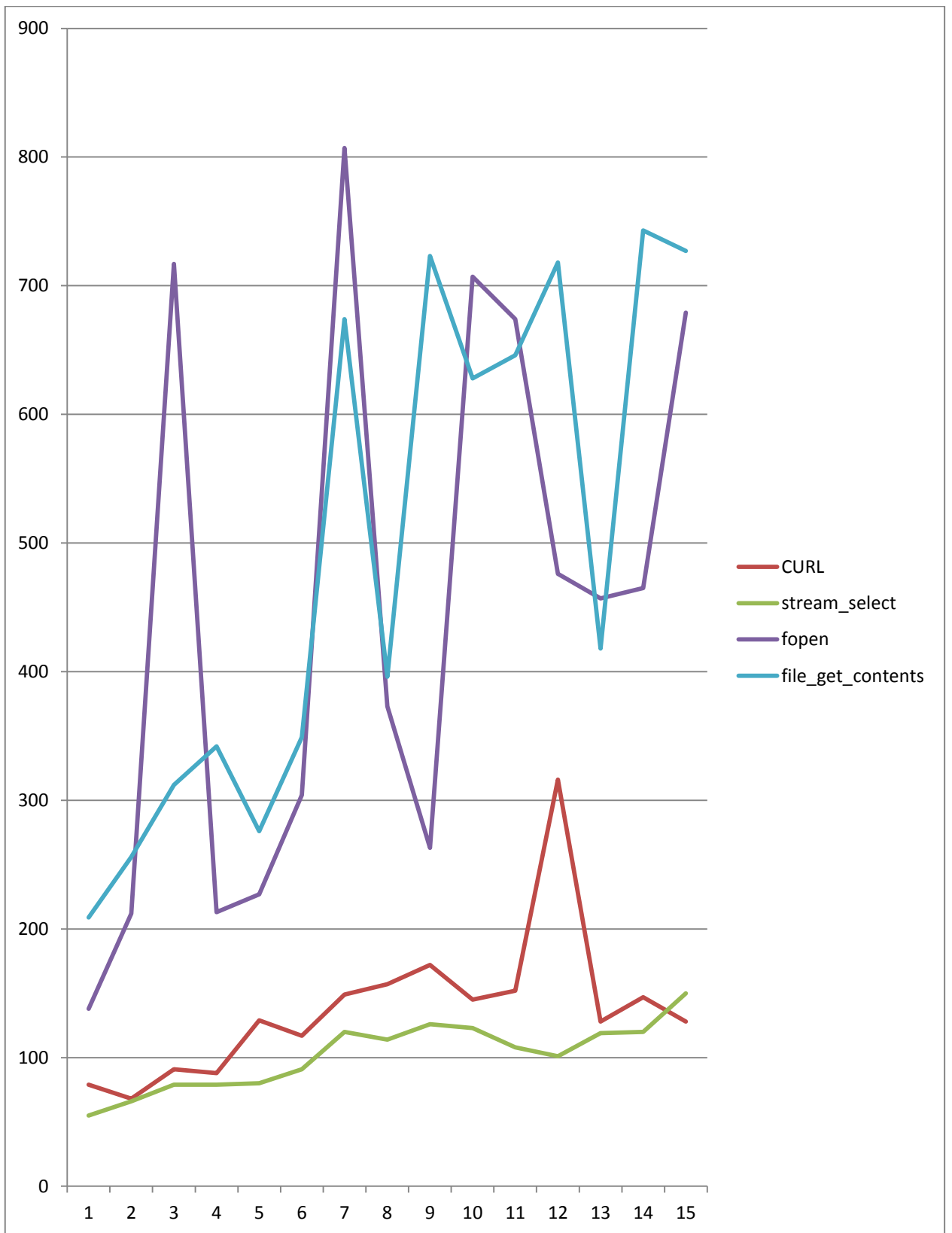
Vėlgi kaip ir pirmuoju atveju situacija panaši, su viena svetaine visi metodai užduotį atlieka per maždaug 30 milisekundžių, o pajungus 10 svetainių, nuoseklūs metodai nuo lygiagrečių atsilieka dvigubai, atitinkamai maždaug 60 milisekundžių ir maždaug 35 milisekundės.

8.2 Laiko pokytis priklausomai nuo reklamų skaičiaus

Matuojant laiką, prie centralizuotos reklamos valdymo sistemos, buvo prijungta 10 svetainių. Į svetaines buvo įkeliama po vienodą skaičių reklamų. Šis tyrimas buvo atliekamas tik dviem funkcijoms, t.y. reklamų perkėlimas ir statistikos surinkimas, nes tik šiais atvejais rezultatai tiesiogiai gali priklausyti nuo reklamų skaičiaus.

Reklamų perkėlimo funkcija turėjo padalinti dešimčiai svetainių, po nustatytą reklamų skaičių, kadangi net ir vienai reklamai perkelti visom svetainėm, reikėjo dešimt sujungimų, tai jau iškarto tapo matomas lygiagrečių metodų pranašumas.

Detali informacija pateikta 5 lentelėje, o rezultatų atvaizdavimas grafiku 6 paveiksle.

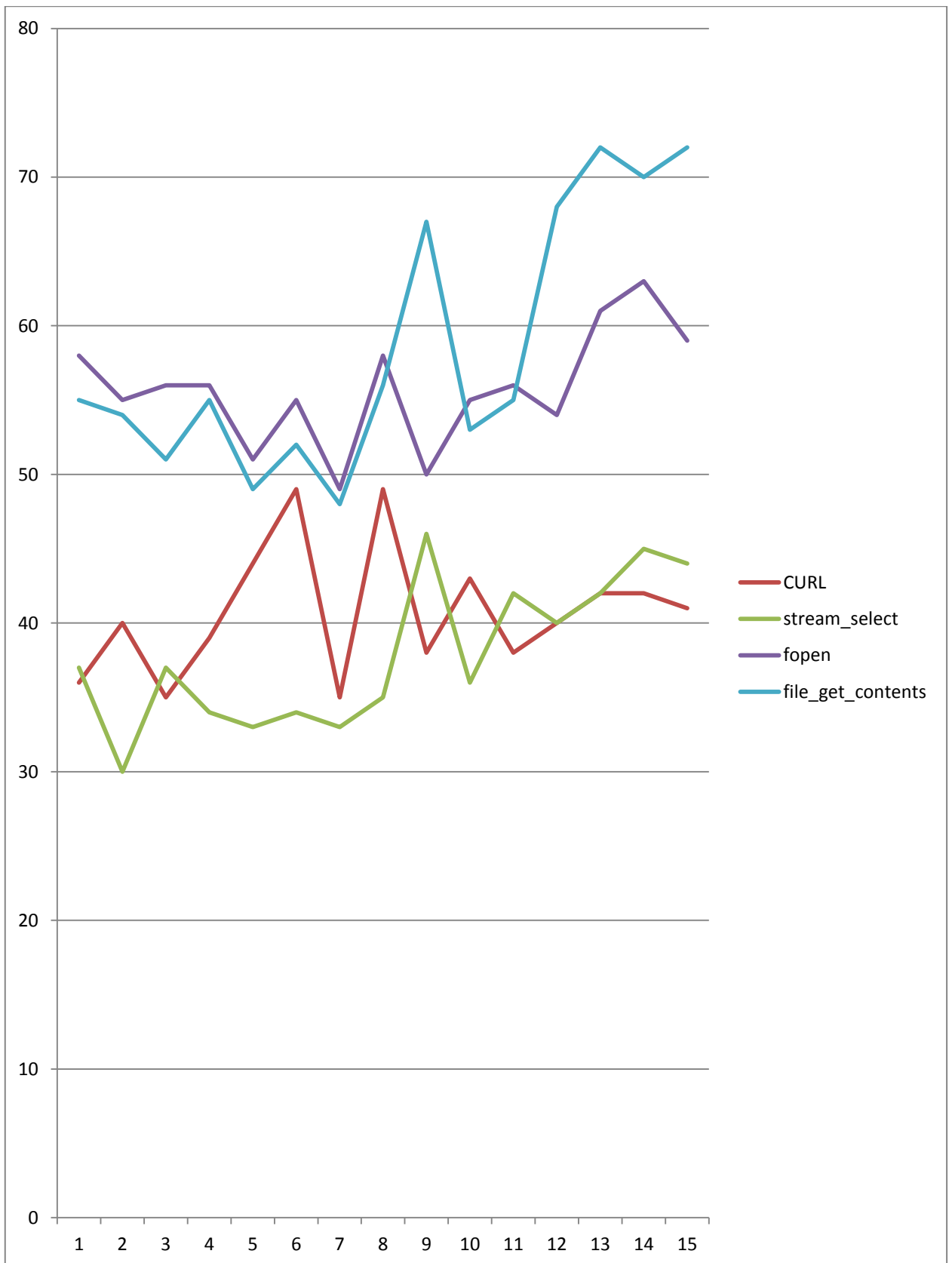


6 Pav. Skirtingų metodų įtaka laikui. Reklamų perkėlimas

Lygiagrečiais metodais atliekama funkcija, kai pajungta dešimt svetainių ir įkeliama po vieną reklamą, savo darbą atliko per maždaug 60 milisekundžių, kai tuo tarpu sujungus nuosekliai užtrunkama maždaug 180 milisekundžių.

Didėjant reklamų skaičiui skirtumas tarp skirtingų metodų tipų tampa vis ryškesnis. Nuosekliai sujungtų svetainių funkcijos atlikimo laikas labai nestabilus, daug didesni svyravimai. Kelis kartus pasiekiamas funkcijos atlikimo laikas kuris yra daugiau nei sekundė. Tai parodo, kad užvėlavus vienam iš serverių, bendras laikas į tai sureaguoja labai juntamai. Tuo tarpu naudojant lygiagretų programavimą, reklamų perkėlimas užtrunka trumpiau, jaučiami mažesni laiko svyravimai.

Padidinus reklamų skaičių iki penkiolikos, matome, kad lygiagrečiais metodais mes pasiekiamo apie 140 milisekundžių laiką, kuris nuo vienos reklamos laiko skiriasi tik dvigubai, nors reklamų skaičius apdidėjęs penkiolika kartų. Nuosekliai sujungtų svetainių funkcijos atlikimo laikas yra apie 700 milisekundžių. Tai yra penkis kartus didesnis nei lyginant su lygiagrečiais metodais. Taipogi laikas stipriai skiriasi ir nuo to kuris buvo perkeliant vieną reklamą. Todėl galima daryti išvadą, kad naudojant nuoseklius metodus mes be didesnių nesklandumų galime valdyti daug mažiau reklamų.



7 Pav. Skirtingų metodų įtaka laikui. Statistikos surinkimas

Statistikos surinkimo metu, pastebima ta pati tendencija. Kaip matome pagal 7 Pav. esantį grafiką, lygiagrečiais metodais sujungtos svetainės rezultatus gražina greičiau. Laikas nuo

reklamų skaičiaus priklauso nežymiai, nes rezultatus mysql serveris gražina pakankamai greitai, kad tai nebūtų juntama.

8.3 Bendras metodų įvertinimas

Naudojami metodai vertinami pagal veikimo greitį, sudėtingumą, panaudojimo galimybes, galimybes pritaikyti kitoms funkcijoms, veikimo patikimumą.

Metodai vertinami nuo 0 iki 10, priklausomai nuo to kiek jie atitinka keliamą reikalavimą, konkrečioje situacijoje.

1 Lentelė. Bendras metodų įvertinimas

	cURL biblioteka	Stream_select	fopen	File_get_contents
Naudingumas	10	10	8	9
Patikimumas	9	9	10	10
Panaudojimo galimybės	9	8	10	10
Veikimo greitis	10	10	5	6
Papildomi reikalavimai	10	9	10	10
Bendras įvertinimas	9.6	9.2	8,6	9

Kaip galima suprasti žiūrint į įvertinimus, lygiagretūs metodai turi pranašumą prieš nuoseklius, nes jie gali užtikrinti žymiai didesnę greitį, sistema veikia stabiliau ir mažiau reaguoja į serverių apkrovimą. Todėl naudojant cURL, arba stream_select metodus darbas vyksta greičiau. Lyginant sudėtingumą, šiuo konkrečiu atveju panaudoti visus keturis metodus nėra sudėtinga, nuosekliam sujungimui naudojami metodai puikiai tinka paprastam duomenų nuskaitymui, o lygiagrečius metodus pritaikyti kitiems skaičiavimams nėra labai paprasta.

9 Išvados

- Atliekant magistrinį darbą, buvo susipažinta su internetinės reklamos rinka, reklamų būdais ir jau egzistuojančiais sprendimais. Sukurti centralizuotą reklamos valdymo sistemą buvo pasirinktas rečiau naudojamas architektūros principas, kad būtų galima detaliau panagrinėti PHP kalbos galimybes, naudoti lygiagrečius programavimo principus.
- Kuriant programą buvo išbandyti ketu metodai, sistemos ir svetainių sujungimui, visi jie veikė be trikdžių ir atliko reikiamas funkcijas. Atliekant bandymus buvo pastebėta, kad serverių apkrova yra sunkiai prognozuojama, todėl bet kuriuo metu vienas ar kitas veiksmas gali užtrukti ilgiau.
- Palyginus sistemos veikimą su nuosekliu vykdymu ir naudojant lygiagrečius metodus, pastebėta, kad:
 - Veikimo laikas naudojant lygiagrečius metodus yra trumpesnis, jeigu yra sujungta daugiau svetainių;
 - Funkcijos atlikimo laikas naudojantis lygiagrečiais principais yra stabilesnis, nes mažiau jaučiamas kurio nors serverio vėlavimas;
- Įvertinus metodų naudingumą, sudėtingumą ir galimybes juos pritaikyti, patogiausia yra naudoti cURL biblioteką, kuri yra nesunkiai pritaikoma ir užtikrina gerus rezultatus. Nuoseklūs metodai tokie kaip fopen ir file_get_contents yra labiau tinkami nuskaityti informacijai iš vieno šaltinio.

10 Literatūra

[1] IBM Develop multitasking applications with PHP V5 [žiūrėta 2010-04-02]

Prieiga per internetą: www.ibm.com/developerworks/web/library/os-php-multitasking

[2] Resolving transaction concurrency issues in a PHP+MSQL multi-user environment

[žiūrėta 2008-11-17]

Prieiga per internetą: www.ardamis.com/2008/04/06/resolving-transaction-concurrency-issues

[3] Carl Patrik Nilsson. *ATTENTION TO ADVERTISING*. 2006.

[4] Bill Appelbe. *DEVELOPING MULTITASKING APPLICATIONS PROGRAMS*. 2008.

[5] Antanas Vidžiūnas, Vytautas Barzdaitis. *Interneto svetainių ir tinklalapių kūrimas*. 2005

[6] Virginijus Rabačius. *ĮMONĖS REKLAMOS INTERNETINĖS SVETAINĖS TURINIO VALDYMO SISTEMA*. 2005

[7] Multitasking in PHP [žiūrėta: 2010-04-02]

Prieiga per internetą: www.jacobsantos.com/2006/general/multitasking-in-php

[8] Stream_select [žiūrėta: 2010-05-01]

Prieiga per internetą: www.php.net/manual/en/function.stream-select.php

[9] curl_multi_getcontent [žiūrėta: 2010-05-01]

Prieiga per internetą: lt2.php.net/manual/en/function.curl-multi-getcontent.php

[10] Guru - multiplexing [žiūrėta: 2010-05-20]

Prieiga per internetą: www.wezfurlong.org/blog/2005/may/guru-multiplexing

[11] How banner ads work [žiūrėta: 2010-05-20]

Prieiga per internetą: www.computer.howstuffworks.com/banner-ad5.htm

[12] Multithreading in PHP with cURL [žiūrėta: 2010-05-20]

Prieiga per internetą: www.ibuildings.nl/blog/archives/811-Multithreading-in-PHP-with-CURL.html

[13] fopen [žiūrēta: 2010-05-20]

Prieiga per internetą: www.php.net/manual/en/function.fopen.php

[14] Selling advertising space [žiūrēta: 2010-05-20]

Prieiga per internetą: www.computer.howstuffworks.com/banner-ad8.htm

[15] Comparing communication speed between different machines in PHP [žiūrēta: 2010-05-20]

Prieiga per internetą: www.frontaufprall.com/2007/08/21/comparing-communication-speed-between-different-machines-in-php/

[16] file_get_contents [žiūrēta: 2010-05-21]

Prieiga per internetą: <http://lt2.php.net/manual/en/function.file-get-contents.php>

11 Terminų ir santrumpų žodynas

PHP – plačiai paplitusi dinaminė interpretuojama programavimo kalba (en: *Hypertext Preprocessor*), sukurta 1997 m. ir specialiai pritaikyta interneto svetainių kūrimui.

JavaScript – objektiškai orientuota programavimo kalba, besiremianti prototipų principu. Dažniausiai kalba naudojama internetinių puslapių interaktyvumo realizacijai, bet taip pat naudojama ir kaip galimybė manipuluoti tam tikromis programomis.

JQuery – javascript biblioteka, supaprastinanti darbą.

AJAX – arba Asinchroninis JavaScript ir XML programavimas – terminas, apibrėžiantis svetainių programavimo technologiją, naudojančią šias priemones maksimaliam interaktyvumui pasiekti:

Dedikuotas serveris – tai serveris, turintis fiksuotą resursų skaičių, kuriais galima manipuluoti.

Virtualus serveris – tai serveris, kuriame yra ir daugiau projektų, visi projektai dalinasi serverio resursais.

cURL – tai biblioteka, kuri suteikia įvairias galimybes duomenų perdavimui, naudojant įvairius protokolus.

Stream_select – funkcija, kuri pagal pateiktą masyvą aktyvių sujungimų, nuskaitymo informaciją. Nuskaitymas vyksta lygiagrečiai.

Fopen – funkcija, kuri atidaro nuskaitymui failą, arba URL adresą.

File_get_contents – funkcija, kuri nuskaitymo informaciją, pagal pateiktą adresą ir gražina *string* tipo kintamąjį.

Subdomenas – tai DNS hierarchija, subdomenas yra domenas kuris yra dalis didesnio domeno.

Gija – tai lygiagretaus programavimo terminas. Gija apibūdina lygiagrečiai vienas šalia kito veikiančius veiksmus. Gijos dalinasi resursais ir atlieka joms pavestas užduotis.

FTP - standartas failų perdavimui. FTP protokolą realizuojanti programinė įranga paprastai vadinama FTP klientais.

FTP yra kliento serverio architektūros protokolas, leidžiantis apsikeisti bet kokio tipo failais be papildomo apdorojimo.

12 Priedai

12.1 Skirtingų metodų įtaka laikui svetainių patikrinimo metu

Skaičiavimai atliekami, kiekvienai svetainei turint po penkias reklamas.

2 Lentelė. Skirtingų metodų įtaka laikui. Svetainių patikrinimas

Metodas Svetainės	cURL	Stresam_select	fopen	File_get_contents
1	32ms 26ms 26ms	21ms 25ms 21ms	30ms 33ms 33ms	27ms 31ms 28ms
	28ms	22ms	32ms	29ms
2	31ms 31ms 29ms	23ms 24ms 28ms	30ms 29ms 28ms	31ms 29ms 35ms
	30ms	25ms	29ms	32ms
3	33ms 27ms 30ms	35ms 37ms 31ms	35ms 29ms 31ms	34ms 35ms 35ms
	30ms	34ms	32ms	34ms
4	35ms 29ms 39ms	36ms 30ms 29ms	33ms 31ms 35ms	33ms 35ms 39ms
	34ms	32ms	33ms	36ms
5	33ms 31ms 37ms	35ms 49ms 47ms	42ms 37ms 38ms	37ms 35ms 47ms
	34ms	44ms	39ms	40ms
6	35ms 29ms 37ms	39ms 31ms 37ms	49ms 41ms 47ms	41ms 55ms 35ms
	34ms	34ms	39ms	44ms
7	39ms 28ms 35ms	37ms 29ms 33ms	43ms 45ms 45ms	61ms 53ms 43ms
	34ms	33ms	43ms	52ms
8	34ms 29ms 40ms	33ms 37ms 35ms	51ms 49ms 53ms	43ms 43ms 39ms
	34ms	35ms	51ms	42ms
9	41ms 31ms 39ms	35ms 31ms 41ms	55ms 58ms 41ms	57ms 51ms 56ms
	37ms	36ms	51ms	55ms
10	35ms 34ms 33ms	63ms 35ms 38ms	56ms 60ms 53ms	55ms 52ms 48ms
	34ms	45ms	56ms	52ms

12.2 Skirtingų metodų įtaka laikui reklamų perkėlimo metu

Skaičiavimai atliekami, kiekvienai svetainei turint po penkias reklamas.

3 Lentelė. Skirtingų metodų įtaka laikui. Reklamų perkėlimas

Metodas Svetainės	cURL	Stresam_select	fopen	File_get_contents
1	67ms 66ms 50ms	82ms 69ms 62ms	45ms 55ms 50ms	55ms 56ms 49ms
	61ms	71ms	50ms	53ms
2	52ms 64ms 55ms	62ms 86ms 53ms	148ms 103ms 76ms	84ms 123ms 101ms
	57ms	67ms	109ms	103ms
3	59ms 60ms 69ms	82ms 72ms 65ms	183ms 111ms 97ms	99ms 82ms 88ms
	63ms	73ms	130ms	90ms
4	99ms 76ms 74ms	55ms 119ms 49ms	174ms 244ms 163ms	398ms 135ms 166ms
	83ms	74ms	194ms	233ms
5	74ms 76ms 88ms	76ms 86ms 76ms	377ms 160ms 115ms	256ms 177ms 184ms
	79ms	79ms	217ms	206ms
6	74ms 67ms 53ms	99ms 130ms 105ms	276ms 424ms 420ms	221ms 175ms 192ms
	65ms	111ms	373ms	196ms
7	151ms 111ms 109ms	86ms 82ms 80ms	312ms 199ms 393ms	356ms 326ms 215ms
	124ms	83ms	301ms	299ms
8	172ms 138ms 98ms	129ms 93ms 103ms	531ms 559ms 228ms	377ms 299ms 191ms
	136ms	108ms	439ms	289ms
9	101ms 164ms 98ms	98ms 98ms 115ms	396ms 233ms 459ms	307ms 357ms 439ms
	121ms	104ms	363ms	368ms
10	101ms 104ms 128ms	129ms 129ms 103ms	636ms 676ms 365ms	619ms 383ms 381ms
	111ms	120ms	559ms	461ms

12.3 Skirtingų metodų įtaka laikui reklamų statistikos surinkimo metu

Skaičiavimai atliekami, kiekvienai svetainei turint po penkias reklamas.

4 Lentelė. Skirtingų metodų įtaka laikui. Statistikos surinkimas

Metodas Svetainės	cURL	Stresam_select	fopen	File_get_contents
1	24ms 25ms 28ms	19ms 26ms 28ms	31ms 22ms 24ms	27ms 26ms 25ms
	26ms	24ms	26ms	26ms
2	27ms 23ms 39ms	31ms 29ms 19ms	30ms 41ms 23ms	34ms 27ms 29ms
	30ms	26ms	31ms	30ms
3	30ms 30ms 24ms	31ms 39ms 31ms	33ms 35ms 47ms	28ms 33ms 33ms
	28ms	34ms	38ms	31ms
4	33ms 37ms 29ms	35ms 35ms 28ms	35ms 37ms 39ms	37ms 31ms 29ms
	33ms	33ms	37ms	32ms
5	45ms 35ms 43ms	33ms 27ms 31ms	43ms 33ms 32ms	39ms 37ms 72ms
	41ms	30ms	36ms	49ms
6	39ms 32ms 30ms	28ms 33ms 43ms	39ms 52ms 52ms	43ms 39ms 39ms
	34ms	38ms	48ms	40ms
7	35ms 34ms 33ms	39ms 39ms 35ms	43ms 49ms 52ms	45ms 45ms 55ms
	34ms	38ms	48ms	48ms
8	35ms 33ms 32ms	35ms 28ms 35ms	45ms 41ms 41ms	46ms 41ms 41ms
	33ms	33ms	42ms	43ms
9	45ms 35ms 33ms	31ms 33ms 31ms	41ms 43ms 51ms	59ms 59ms 51ms
	38ms	32ms	45ms	56ms
10	37ms 41ms 32ms	32ms 39ms 41ms	56ms 49ms 45ms	66ms 58ms 55ms
	37ms	37ms	50ms	60ms

12.4 Skirtingų metodų įtaka laikui reklamų perkėlimo metu, kai kinta reklamų skaičius

Skaičiavimai atliekami, kai prijungta dešimt reklamas publikuojančių svetainių..

5 Lentelė. Skirtingų metodų įtaka laikui. Reklamų perkėlimas

Metodas Reklamos	cURL	Stresam_select	fopen	File_get_contents
1	62ms 64ms 112ms	64ms 53ms 49ms	146ms 145ms 124ms	326ms 158ms 143ms
	79ms	55ms	138ms	209ms
2	87ms 80ms 66ms	69ms 65ms 65ms	227ms 201ms 209ms	367ms 210ms 192ms
	68ms	66ms	212ms	256ms
3	97ms 95ms 80ms	72ms 84ms 82ms	1190ms 508ms 453ms	348ms 189ms 399ms
	91ms	79ms	717ms	312ms
4	82ms 78ms 103ms	77ms 86ms 74ms	189ms 236ms 213ms	551ms 229ms 245ms
	88ms	79ms	213ms	342ms
5	152ms 158 78ms	74ms 71ms 94ms	262ms 195ms 225ms	304ms 240ms 285ms
	129ms	80ms	227ms	276ms
6	135ms 97ms 119ms	68ms 103ms 103ms	221ms 221ms 469ms	508ms 358ms 182ms
	117ms	91ms	304ms	349ms
7	161ms 150ms 137ms	116ms 138ms 106ms	721ms 782ms 918ms	697ms 664ms 662ms
	149ms	120ms	807ms	674ms
8	156ms 172ms 144ms	108ms 86ms 149ms	588ms 219ms 313ms	486ms 269ms 432ms
	157ms	114ms	373ms	396ms
9	168ms 152ms 195ms	119ms 114ms 145ms	260ms 314ms 216ms	1420ms 471ms 279ms
	172ms	126ms	263ms	723ms
10	100ms 152ms 183ms	105ms 133ms 131ms	1220ms 504ms 396ms	1240ms 377ms 268ms
	145ms	123ms	707ms	628ms
11	135ms 164ms 156ms	140ms 84ms 100ms	668ms 957ms 397ms	548ms 1100ms 291ms
	152ms	108ms	674ms	646ms
12	537ms 225ms 186ms	100ms 119ms 84ms	479ms 324ms 625ms	418ms 740ms 994ms
	316ms	101ms	476ms	718ms
13	113ms 174ms 96ms	88ms 120ms 149ms	379ms 590ms 401ms	403ms 504ms 348ms
	128ms	119ms	457ms	418ms
14	167ms 128ms 146ms	122ms 131ms 108ms	499ms 506ms 391ms	1260ms 469ms 501ms
	147ms	120ms	465ms	743ms
15	148ms 137ms 100ms	167ms 120ms 163ms	587ms 469ms 981ms	694ms 1080ms 408ms
	128ms	150ms	679ms	727ms

12.5 Skirtingų metodų įtaka laikui statistikos surinkimo metu, kai kinta reklamų skaičius

Skaičiavimai atliekami, kai prijungta dešimt reklamas publikuojančių svetainių..

6 Lentelė. Skirtingų metodų įtaka laikui. Statistikos surinkimas

Metodas Reklamos	cURL	Stresam_select	fopen	File_get_contents
1	33ms 37ms 37ms	45ms 34ms 31ms	49ms 65ms 59ms	47ms 49ms 69ms
	36ms	37ms	58ms	55ms
2	41ms 43ms 35ms	29ms 29ms 33ms	58ms 55ms 52ms	58ms 58ms 45ms
	40ms	30ms	55ms	54ms
3	37ms 35ms 33ms	38ms 37ms 35ms	55ms 58ms 55ms	51ms 49ms 55ms
	35ms	37ms	56ms	51ms
4	39ms 43ms 35ms	39ms 32ms 31ms	60ms 60ms 47ms	51ms 68ms 47ms
	39ms	34ms	56ms	51ms
5	43ms 54ms 35ms	33ms 32ms 33ms	49ms 56ms 49ms	52ms 51ms 45ms
	44ms	33ms	51ms	49ms
6	45ms 43ms 59ms	31ms 37ms 33ms	48ms 62ms 55ms	47ms 56ms 53ms
	49ms	34ms	55ms	52ms
7	37ms 35ms 34ms	35ms 36ms 29ms	53ms 45ms 49ms	49ms 43ms 53ms
	35ms	33ms	49ms	48ms
8	72ms 41ms 35ms	34ms 41ms 31ms	67ms 52ms 56ms	55ms 57ms 55ms
	49ms	35ms	58ms	56ms
9	43ms 39ms 31ms	43ms 54ms 42ms	47ms 55ms 47ms	56ms 86ms 59ms
	38ms	46ms	50ms	67ms
10	51ms 34ms 45ms	37ms 31ms 41ms	57ms 54ms 53ms	55ms 56ms 47ms
	43ms	36ms	55ms	53ms
11	37ms 37ms 41ms	41ms 41ms 43ms	55ms 55ms 57ms	59ms 49ms 57ms
	38ms	42ms	56ms	55ms
12	45ms 37ms 37ms	45ms 37ms 39ms	51ms 58ms 52ms	70ms 72ms 63ms
	40ms	40ms	54ms	68ms
13	45ms 39ms 42ms	47ms 37ms 41ms	55ms 59ms 69ms	71ms 64ms 82ms
	42ms	42ms	61ms	72ms
14	37ms 51ms 38ms	41ms 56ms 38ms	59ms 59ms 72ms	64ms 64ms 81ms
	42ms	45ms	63ms	70ms
15	39ms 44ms 41ms	36ms 45ms 52ms	63ms 55ms 58ms	72ms 78ms 65ms
	41ms	44ms	59ms	72ms