



KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Vytautas Jankevičius

**ATVIROJO KODO DUOMENŲ BAZIŲ SERVERIŲ
ANALIZĖ APTARNAVIMO PASLAUGŲ
TEIKIMO VALDYMO SISTEMOS PAGRINDU**

Magistro darbas

Kalbos konsultantė

lekt. dr. Ilona Mickienė

2006-05-

Vadovas

doc. dr. R. Butleris

2006-05

Recenzentas

doc. Dr. V.

Pilkauskas

2006-05-

Atliko

IFM 0/2 gr. stud.

Vytautas Jankevičius

2006-05-

Kaunas, 2006

Turinys

Lentelių sąrašas	2
Paveikslėlių sąrašas	2
Įvadas	3
1. Panašių sistemų ir atvirojo kodo duomenų bazių serverių analizė.....	5
1.1. Aptarnavimo paslaugų teikimo valdymo sistemų funkcinių savybių analizė	5
1.1.1. Įmonėje „GNT Lietuva“ naudojama aptarnavimo paslaugų teikimo valdymo sistema..	5
1.1.2. Įmonėje ACME naudojama aptarnavimo paslaugų teikimo valdymo sistema.....	6
1.1.3. Įmonėje Servicenet naudojama aptarnavimo paslaugų teikimo valdymo sistema.....	7
1.1.4. Įmonėje ELKO naudojama aptarnavimo paslaugų teikimo valdymo sistema.....	7
1.2. Aptarnavimo paslaugų teikimo valdymo sistemų palyginimas.....	8
1.2. Atvirojo kodo duomenų bazių serverių analizė	9
1.2.1. MySQL – duomenų bazės serveris.....	9
1.2.2. PostgreSQL – duomenų bazės serveris	12
1.2.3. Firebird – duomenų bazės serveris.....	14
1.2.4. Atvirojo kodo duomenų bazių serverių panašumai ir skirtumai	15
1.3. Vertinimo kriterijai	16
2. Aptarnavimo paslaugų teikimo valdymo sistemos projekto modelis	18
2.1. Sistemos panaudos atvejai.....	18
2.2. Duomenų bazė	19
2.3. Sistemos loginė architektūra.....	19
2.3.1. Paketo „Vartotojo Sąsaja“ klasė ir metodai.....	20
2.3.2. Paketo „Veiksmai“ klasės ir metodai	21
2.3.3. Paketo „Tyrimas“ klasė ir metodas	22
2.4. Dinaminis sistemos vaizdas.....	24
2.4.1. Veiksmų eiliškumo diagramos panaudos atvejams.....	24
2.4.2. Sąveikos diagramos.....	28
3. Atvirojo kodo duomenų bazių serverių tyrimo rezultatai	31
3.1. Įterpimo funkcijos tyrimas	31
3.2. Koregavimo funkcijos tyrimas	34
3.3. Paieškos funkcijos tyrimas	37
3.4. Trynimo funkcijos tyrimas	40
3.5. Tyrimo rezultatų suvestinė	42
Santrauka (anglų kalba)	45
Terminų ir santrmupų žodynas.....	46
Išvados	47
Literatūra.....	49
Priedai	50

Lentelių sąrašas

1 lentelė. Panašių sistemų palyginimas	9
2 lentelė. Duomenų bazių serverių palyginimas	16
3 lentelė. Sąveikos diagramos	28

Paveikslėlių sąrašas

1 pav. Panaudos atvejų diagrama	18
2 pav. Duomenų bazės esybių ryšių diagrama	19
3 pav. Sistemos loginė architektūra.....	20
4 pav. Vartotojo sąsajos klasė.	20
5 pav. Paketo „Veiksmi“ klasių diagrama.....	21
6 pav. Klasė „Tyrimas“ ir jos metodai.....	24
7 pav. Veiksmų eiliškumo diagrama vartotojų sukūrimui/redagavimui/naikinimui.....	25
8 pav. Veiksmų eiliškumo diagrama informacijai apie klientą administruoti	25
9 pav. Veiksmų eiliškumo diagrama prekių statusui peržiūrėti	25
10 pav. Veiksmų eiliškumo diagrama žinutėms siusti/gauti.....	26
11 pav. Veiksmų eiliškumo diagrama ataskaitoms peržiūrėti.....	26
12 pav. Veiksmų eiliškumo diagrama prekių gedimui administruoti.....	26
13 pav. Veiksmų eiliškumo diagrama darbų grafikui	27
14 pav. Veiksmų eiliškumo diagrama naujoms prekėms administruoti.....	27
15 pav. Veiksmų eiliškumo diagrama iškvietimą registruoti.....	27
16 pav. Veiksmų eiliškumo diagrama iškvietimų sąrašui su adresu žemėlapyje.....	28
17 pav. Įterpimas po 50 tūkst. eilučių.....	32
18 pav. Įterpimas po 100 tūkst. eilučių.....	32
19 pav. Skirtingo dydžio duomenų įterpimas	33
20 pav. Įrašų skaičiaus ir duomenų bazės dydžio priklausomybė.....	34
21 pav. Koregavimas, nurodant ID.....	35
22 pav. Koregavimas, nurodant visą lauką	36
23 pav. Koregavimas, nurodant fragmentą.....	36
24 pav. Koregavimas, nurodant ID, visą lauką ir fragmentą	37
25 pav. Paieška, nurodant visą lauką.....	38
26 pav. Paieška, nurodant fragmentą.....	39
27 pav. Paieška, nurodant visą lauką ir fragmentą	39
28 pav. Trynimas, nurodant visą lauką.....	40
29 pav. Trynimas, nurodant fragmentą.....	41
30 pav. Trynimas, nurodant visą lauką ir fragmentą.....	42
31 pav. Rezultatų suvestinė.....	43
32 pav. Rezultatų suvestinė, kai lentelės dydis neviršija 500 000 įrašų.....	44

Ivadas

Techninė įranga sparčiai tobulėja, procesorių taktiniai dažniai didėja, operatyvioji atmintis tiek spartėja, tiek didėja. Nors kietųjų diskų talpos auga, jų sparta didėja labai lėtai, todėl pasirinkti tinkamą duomenų saugojimo variklį yra labai svarbu. Yra žinoma, kad sistemos su duomenų bazėmis greičiau pasiekia, saugo ar atlieka kitokius veiksmus su duomenimis, nei sistemos saugančios duomenis paprastoje failų sistemoje. Sistemos greitį veikia daug faktorių: sistemos realizavimo tipas, programavimo kalba, optimalus programos realizavimas bei duomenų bazės serveris.

Duomenų bazių serverius kuria tokios didžios organizacijos kaip Microsoft, IBM, Oracle, tačiau yra ir kitų – nepelno siekiančių organizacijų, kurios savo produktais leidžia naudotis nemokamai. Internete galima rasti galybę atvirojo kodo duomenų bazių serverių, tačiau populiariausi yra trys: MySQL, PostgreSQL ir Firebird.

Temos aktualumas. Aptarnavimo paslaugų teikimo valdymo sistemai, kaip ir kitoms per interneto prieigą pasiekiamoms programų sistemoms, svarbus operacijų įvykdymo greitis. Pati programų sistema nepasiekia duomenų, už tai atsakingas duomenų bazės serveris, ji paprasčiausiai reikiamos informacijos iš duomenų bazės serverio Daugumoje atveju programų sistema veiksmus su duomenimis atlieka daug greičiau, nei duomenų bazės serveris sugeba atlikti pateiktus darbus. Lėtai veikianti programų sistema vartotojo lūkesčių nepatenkins. Taigi tinkamo duomenų bazės serverio pasirinkimas yra gana svarbus.

Temos pasirinkimo motyvas. Aptarnavimo paslaugų teikimo valdymo sistema naudoja MySQL duomenų bazės serverį, kuris buvo pasirinktas atlikus atvirojo kodo duomenų bazių serverių analizę. Šiuo tyrimu nustatyta ar pasirinktas duomenų bazės serveris optimalus aptarnavimo paslaugų teikimo valdymo sistemai.

Darbo objektas – atvirojo kodo duomenų bazės serverių analizė.

Tyrimo hipotezė – atvirojo kodo duomenų bazių serverių savybės yra labai panašios, skiriasi tik kelios funkcijos, arba jos vadinamos skirtingais vardais nors atlieka tą pačią funkciją. Įvairių operacijų atlikimo greičiai yra labai panašūs, tačiau jie negali būti identiški.

Darbo tikslas yra ištirti populiariausius atvirojo kodo duomenų bazės serverius ir nustatyti, kuris padės aptarnavimo paslaugų teikimo valdymo sistemai dirbti greičiau.

Tyrimo uždaviniai :

- Atlikti teorinę atvirojo kodo duomenų bazių serverių analizę.
- Išskirti atvirojo kodo duomenų bazių serverių panašumus ir skirtumus.
- Išanalizuoti aptarnavimo paslaugų teikimo valdymo sistemų funkcines savybes.
- Palyginti aptarnavimo paslaugų teikimo valdymo sistemų funkcines savybes.
- Suprojektuoti specialią tyrimams reikalingą klasę bei metodus.

- Naudojantis sukurta klase atlikti tyrimus bei nustatyti kuris duomenų bazių serveris optimaliausiai tinka aptarnavimo paslaugų teikimo valdymo sistemai.

Tyrimo metodika: atliekant magistro tyrimą buvo naudotasi specialiai testavimams sukurta klase ir metodais. Sukurtų metodų pagalba buvo atlikti tyrimai registruojant rezultatus, kuriuos grafiškai apdorojus nustatomas optimalus aptarnavimo paslaugų teikimo valdymo sistemos duomenų bazės serveris.

Darbo struktūra: magistro darbą sudaro trys pagrindinės dalys: panašių sistemų ir atvirojo kodo duomenų bazių serverių analizė, aptarnavimo paslaugų teikimo valdymo sistemos projekto modelis ir atvirojo kodo duomenų bazių serverių tyrimo rezultatai.

Pirmoje dalyje pateikta aptarnavimo paslaugų teikimo valdymo sistemų funkcinių savybių apžvalga bei palyginimai, taip pat atvirojo kodo duomenų bazių serverių apžvalga bei palyginimas.

Antroje dalyje pateikiamas aptarnavimo paslaugų teikimo valdymo sistemos projekto modelis t. y. panaudos atvejų diagrama, duomenų bazės schema, sistemos loginė schema, klasių diagramos, veiksmų eiliškumo diagramos bei sveikos diagramos.

Trečioji dalyje pateiktas atvirojo kodo duomenų bazių serverių tyrimo rezultatai, t. y. įvairūs grafikai vaizduojantys visų tirtų funkcijų tyrimų rezultatus.

Darbo apimtis ir naudoti šaltiniai: magistro darbą sudaro 49 lapai, pirmąją darbo dalį sudaro 13 lapų, antrąją 13 lapų ir trečiąją dalį sudaro 14 lapų. Baigiamajame darbe yra 3 lentelės bei 32 paveikslai.

1. Panašių sistemų ir atvirojo kodo duomenų bazių serverių analizė

Aptarnavimo paslaugų teikimo valdymo sistema buvo kuriama ir dokumentuojama pusantrų metų, darbo metu buvo išsiaiškinti reikalavimai sistemai, atlikta naudojamų įrankių analizė, pasirinkti įrankiai, bei užbaigta sistema.

Šiuo darbu apžvelgiamos panašios sistemos, jų funkcijos, pasirinkti įrankiai, po to jie lyginami. Ypatingas dėmesys atkreipiamas į duomenų bazių serverius, kadangi tokiose sistemose reikia greito duomenų pateikimo, o darbas su kietuoju disku yra lėčiausias procesas, palyginus su kitomis sudedamosiomis dalimis.

1.1. Aptarnavimo paslaugų teikimo valdymo sistemų funkcinių savybių analizė

Pagal pagrindinį pritaikymą ir funkcionalumą panašiausios yra kompiuterinės technikos tiekėjų sistemos, bei oficialūs garsių kompiuterinės techniko gamintojų garantiniai centrai.

1.1.1. Įmonėje „GNT Lietuva“ naudojama aptarnavimo paslaugų teikimo valdymo sistema

GNT Lietuva tiekia rinkai kompiuterinę, programinę įrangą, kompiuterinius žaidimus ir įrangą jiems, periferijos įrenginius bei buitines elektronikos prekes. Kompanijos tikslas – siūlyti kruopščiai atrinktą produktų spektrą, naujausias technologijas, greičiausius sprendimus. Tai padidintų klientų darbo sėkmę ir užtikrintų jų nuolatinę pelningą veiklą. Visas dėmesys skiriamas klientų aptarnavimo kokybei, užsakymo ir pristatymo paprastumui, pasitelkiamos moderniausios interneto technologijos.[4]

GNT interneto puslapis naudojamas prekėms parduoti bei garantiniam prekių remontui registruoti. Garantinio remonto registravimo pakete vyrauja mėlyna ir balta spalvos, šių spalvų derinys puikiai dera ir neblaško vartotojo dėmesio. Kiekvieno lango viršuje visada matomas įmonės pavadinimas bei įmonės atstovo vardas ir pavardė. Atsidarius garantinio remonto paketą, matomas keletą paskutinių remontų sąrašą. Čia matomas registracijos numeris, prekės būseną, data, kada prekė perėjo į esamą būseną, prekes pavadinimas bei pastaba. Puslapyje yra vartotojo vadovas, tai palengvina naujų vartotojų dalią. Šalia yra informacija apie atliktus pakeitimus sistemoje.

Atsidarius prekės gedimo registravimo langą, matomas vartotojo vadovas, informacija apie klientą. Viskas pildoma viename lange, nurodomas transporto tipas, data bei laikas, kada prekė bus paruošta siuntimui į garantinį centrą. Siuntos paėmimo ir grąžinimo adresai nurodomi tame pačiame lange. Kiek žemiau yra prekės aprašymo laukai. Nurodoma prekės serijos numeris, prekės pavadinimas, gedimas bei pastaba. GNT automatiškai tikrina, ar prekė pirka iš jų ir ar ji yra garantinė. Jei viskas teisingai užpildyta, ekrane matoma visa informacija apie

prekę, jos gedimą ir jos transportavimo ypatybes. Registracijos aktas yra spausdinamas ir pridedamas prie prekės.

Prekės būsenas pildo GNT darbuotojai, taip prekės būseną yra lanksti ir teikia daug informacijos.

1.1.2. Įmonėje ACME naudojama aptarnavimo paslaugų teikimo valdymo sistema

ACME įsikūrė 1993 metais, veiklą pradėjo nuo garso bei vaizdo kasečių prekybos. Šiuo metu stipriai išsiplėtusi įmonė užsiima didmenine prekyba kompiuterių dalimis, informacijos laikmenomis bei žaliavomis joms.[5]

ACME turi interneto puslapį, kuriame parduoda prekes. Jame yra integruotas garantinio aptarnavimo paketas. Šis paketas yra gana dažnai atnaujinamas, tai rodo pastovų įmonės domėjimąsi kliento patogumu.

Puslapyje vyrauja mėlyna ir balta spalvos, kurios gerai dera ir neerzina vartotojo akių. Puslapio viršuje visada matoma informacija apie klientą, vardas, pavardė, įmonė, kurią jis atstovauja.

Atsidarius garantinio aptarnavimo paketą, ekrane matomas paskutinių prekių sąrašas, suskirstytas į būsenų grupes, taip klientui patogiu sekti prekės judėjimą. Puslapyje yra laukas sugedusios prekės serijos numeriui suvesti, pagal jį prekės automatiškai ieškoma duomenų bazėje. Jei serijos numeris nebuvo aptiktas duomenų bazėje, siūloma pasirinkti prekės tipą, priklausomai nuo prekės, ji gali būti iki 4 tipų. Pasirinkus visus tipus, rodoma lentelė, kurioje nurodomas serijos numeris, prekės modelis, komplektacija, bei gedimas. ACME turi gedimų sąrašą, tad nereikia pačiam vartotojui aprašyti gedimo, nebent nebūtų rasta norimo iš pasiūlytų. Paspaudus patvirtinimo mygtuką, gedimas užregistruojamas. Vienas iš paskutinių etapų – transporto pasirinkimas. ACME siūlo du variantus – kliento transportą arba ACME užsakytus kurjerius, už kuriuos teks sumokėti klientui. Paskutinysis etapas – tai prekės registravimo akto, kurį reikia pridėti prie siunčiamos prekės, atspausdinimas. Užregistruota prekė atsiras bendrame kliento sugedusių prekių sąraše su būseną „Išsiųsta į servisą“. ACME naudoja penkias prekės būsenas: „išsiųsta į servisą“, „servisas gavo“, „remontas baigtas“, „išsiųsta klientui“, bei „klientas gavo“. Atsidarius langą apie sugedusią prekę, matoma visa susijusi su preke informacija, t. y. akto numeris, informacija apie klientą, prekės pavadinimas, jos serijos numeris, gedimo aprašymas bei techniko pastaba. Taip pat matoma, kaip prekė keitė būsenas, matoma data, kada prekė pakeitė būseną, pastabos ir darbuotojas, kuris atliko su būsenos kitimu susijusius veiksmus.

1.1.3. Įmonėje Servicenet naudojama aptarnavimo paslaugų teikimo valdymo sistema

ServiceNet LT yra autorizuotas HP, Compaq, IBM, Epson, Samsung, Hyundai, Philips, NEC, Toshiba, Hitachi, Fujitsu-Siemens, Sampo, Belinea, Acer, LG, BenQ, Lexmark įrangos serviso centras.[7]

Interneto puslapyje yra registruojamos sugedusios prekės – tiek garantinės, tiek po garantinės. Puslapyje vyrauja mėlyna ir balta spalvos, šis spalvų derinys malonus akiai, nevargina vartotojo ir neblaško jo dėmesio.

Lango viršuje visada matomas įmonės pavadinimas kartu su ją atstovaujančio asmens vardu bei pavarde. Pagrindiniame lange pateikiamas sugedusių prekių sąrašas, kuriame pateikiamas akto numeris, sugedusios prekės būseną, būsenos pasikeitimo data, prekės aprašymas, remonto tipas bei pastaba. Paspaudus pele ant norimo įrašo, ekrane atidaromas langas. Jame pateikiama visa su sugedusia preke susijusi informacija: prekės pavadinimas, jos serijos numeris, gedimas, transportavimo savybės, kliento pavadinimas, atsakingas asmuo.

Norint registruoti prekės gedimą, reikia spustelti ant nuorodos „Naujas“, tada ekrane suformuojamas prekės registravimo langas. Su klientu susijusi informacija užpildoma automatiškai, reikia pasirinkti, kaip prekė bus pristatyta į serviso centrą – kliento transportu arba per kurjerius, kuriuos užsakys Servicenet. Informacijos apie prekę laukuose reikia nurodyti prekes gamintoją (pasirenkamas iš sąrašo), prekės pavadinimą, gedimą, remonto tipą, t. y. ar prekė garantinė. Užpildžius laukus, spaudžiamas registravimo mygtukas, sistema tikrina, ar laukai užpildyti korektiškai, jei taip, ji parodo visą su preke susijusią informaciją bei siūlo atspausdinti aktą, kuris turi būti pridėtas prie prekės.

1.1.4 Įmonėje ELKO naudojama aptarnavimo paslaugų teikimo valdymo sistema

UAB „ELKO Kaunas“ veiklos sritis yra didmeninė prekyba kompiuterine bei programine įranga. Bendra Lietuvos Latvijos įmonė UAB „ELKO Kaunas“ buvo įkurta 1994 m. Tai pirmoji Lietuvoje paskirstymo kompanija, pradėjusi prekiauti kompiuterių komponentais.[6]

Interneto puslapyje integruota elektroninė parduotuvė su garantinių prekių aptarnavimu. Puslapyje vyrauja mėlyna ir baltos spalvos, kurių atspalviai nedera tarpusavyje, ryški mėlyna spalva rėžia akį, o baltos spalvos tekstas sunkiai skaitomas, langai grubūs, neišnaudojamas visas ekrano plotas nuteikia nemaloniai.

Pirmajame puslapyje matomas gedimo registravimo langas. Jame reikia įvesti prekės serijos numerį, sąskaitos faktūros, kurios pagrindu buvo pirktą prekė, numerį, prekės pavadinimą bei gedimą. Naujos prekės pirmoji būseną - tikrinami serijos ir sąskaitos faktūros numeriai, pateiktų duomenų korektiškumą kontroliuoja žmogus. Nei vienas iš laukų nepildomas

automatiškai, nėra pasirinkimo iš sąrašo galimybės. Paspaudus registracijos mygtuką, ekrane pateikiamas visų sugedusių prekių sąrašas, kuriame matomas registracijos numeris, prekės serijos numeris, pavadinimas, gedimo aprašymas, atliktos operacijos/būsena, data, kada būsena keitėsi, pastabos bei nuoroda į anketą, kurią reikia atspausdinti ir pridėti prie prekės. Sąrašą kontroliuoja pats klientas, t. y. jis sprendžia, ar įrašas yra jam reikalingas. Jei įrašas jam nebereikalingas, jį galima ištrinti.

1.2 Aptarnavimo paslaugų teikimo valdymo sistemų palyginimas

Įmonės interneto puslapis yra labai svarbus, ypač įmonėms, kurios teikia paslaugas internetu. Aptartos keturios sistemos savo funkcinėmis galimybėmis labai panašios, tačiau skirtumų yra. Pirmiausia, atsidarius interneto puslapį, yra spalvos ir langų išdėstymas. GNT, ACME, Servicenet interneto puslapiuose puikiai suderintos spalvos ir langų dydžiai, puslapiai nėra išsukiantys, juose nesunkiai skaitomas tekstas. Tačiau ELKO interneto puslapio spalvų derinys išsukiantis, per ryškus fonas, baltas tekstas sunkiai perskaitomas, langai ir lentelės nekorektiškos, primenančios DOS terpės sistemų vartotojo sąsajas.

Pagrindinis (pirmasis, įjungus sistemą) puslapis ELKO sistemoje yra naujo gedimo registravimas, likusios aptartos sistemos ekrane formuoja esamų/atliktų darbų sąrašą.

Registruojant prekes, ELKO, Servicenet ir GNT naudoja vieną langą, tuo tarpu ACME – tris langus. Viena suvedamas prekės serijos numeris, kitame informacija apie prekę, trečiasis skirtas transportavimo būdui nusakyti.

Prekės pristatymą galima pasirinkti trijose sistemose: GNT, ACME ir Servicenet. Pasirinkimui naudojami du variantai: kliento transportas ar įmonės užsakytas kurjeris. Abiem atvejais klientas apmoka transporto išlaidas. GNT prašo nurodyti, kada kurjeris turėtų atvykti, likusios sistemos atsiunčia kurjerį kiek įmanoma greičiau.

Vienintelis ELKO reikalauja nurodyti sąskaitos faktūros, kurios pagrindu buvo pirktą prekę, numerį, kitose sistemose tai nurodomai tik tuo atveju, kai prekės nerandamos duomenų bazėje.

Vienintelis ELKO nesinaudoja automatizuota paieška duomenų bazėje pagal serijos numerį. Tik ELKO leidžia trinti įrašus iš atliktų prekių taisymo sąrašo. Kitų sistemų sąrašą tvarko įmonės darbuotojai, kartas nuo karto ištrindami atliktų taisymų įrašus.

Vienintelis ACME naudojami penkiomis fiksuotomis prekių būsenomis, kitų sistemų būsenos nurodomos įmonės darbuotojų.

ACME sąrašas suskirstytas į grupes pagal esamas būsenas, panaudotas medžio tipas. Kitos sistemos rodo visų būsenų sąrašą. GNT ir ServiceNet siūlo rūšiavimo būdus pagal datą, būseną, pavadinimą.

	GNT	ACME	Servicenet	ELKO
Spalvos	Dera	Dera	Dera	Nedera
Kurjerio paslaugos	Siūloma	Siūloma	Siūloma	Nesiūloma
Prekių tipai	Nėra	Iki 4 lygių	Nėra	Nėra
Paieška pagal serijos numerį	Yra	Yra	Yra	Nėra
Langų sk. registruojant gedimą	Vienas	Trys	vienas	Vienas
Pirmasis puslapis	Prekių sąrašas	Prekių sąrašas	Prekių sąrašas	Gedimo registracija
Prekių suskirstymas į grupes	Tik rūšiavimas	Suskirstyta pagal būsenas	Tik rūšiavimas	Tik rūšiavimas
Prekių sąrašo koregavimas	Rūšiavimo pasirinkimas	Pasirinkimas, kurias būsenas rodyti	Rūšiavimo pasirinkimas	Galima ištrinti atliktų darbų įrašus
Vartotojo vadovai	Yra	Yra	Yra	Nėra
Komentariai	Tik įmonės darbuotojų	Tik įmonės darbuotojų	Tik įmonės darbuotojų	Tik įmonės darbuotojų

Šaltinis: sukurta autoriaus.

1.2. Atvirojo kodo duomenų bazių serverių analizė

Daugelių atvejų programinės įrangos efektyvumas priklauso nuo to, kaip greitai yra pasiekiami duomenys. Darbui su duomenimis dažniausiai naudojamas duomenų bazės serveris, nuo kurio tiesiogiai priklauso sistemos darbo su duomenimis, t. y. įterpimo, paieškos, koregavimo bei trynimo operacijų, greitis. Kadangi mokami duomenų bazių serveriai kainuoja pakankamai daug, straipsnyje nagrinėjami atvirojo kodo duomenų bazių serveriai. Atkreiptas dėmesys į architektūrą, valdymą ir transakcijas.

1.2.1. MySQL – duomenų bazės serveris

MySQL – populiariausia pasaulyje atvirojo kodo duomenų bazių valdymo sistema. Pagal kai kuriuos kriterijus, tokius kaip aplikacijos raida, duomenų integravimas ar valdymas, MySQL duomenų bazių serveris lyginasi su garsiausiomis mokamomis duomenų bazių serverių sistemomis. Todėl daugybė organizacijų, mokymo įstaigų, Vyriausybės institucijos pasitiki MySQL ir patiki jam savo duomenis. Pasaulyje yra daugiau nei aštuoni milijonai įdiegtų ir aktyviai dirbančių MySQL kopijų.[1]

MySQL pasirinko tokie interneto puslapiai kaip Yahoo, Google, Friendster ir kiti, kurie apkrauti dideliais judėjimais. Kaip kad Los Alamos Lab išbandė MySQL terabaito dydžio duomenų saugykloms valdyti.

Egzistuoja trys MySQL prioritetai:

- Patikimumas. Nereikalauja priežiūros tam, kad sistema veiktų.
- Charakteristikos. Aukštos charakteristikos ir trumpas reakcijos laikas visoms duomenų bazės operacijoms.
- Paprastumas. Instaliacija ir konfigūracija neužtrunka daugiau nei 15 min ir reikalauja labai nedaug nuolatinės priežiūros ar tradicinių administravimo darbų.

ARCHITEKTŪRA

Lankstusis modelis. MySQL yra idealiai pritaikytas dirbti su Intel, AMD ir PowerPC serverių aplinkomis, kurios palaiko terabaito dydžio duomenų bazių saugyklas. Daugybė MySQL klientų naudoja MySQL dėl nepriklausomos architektūros, tai leidžia įsigyti palyginus pigią techninę įrangą.

Didelių duomenų bazių palaikymas. Duomenų dydžio atžvilgiu MySQL praktiškai neturi ribos, duomenų saugyklos dydis gali siekti ir 64 TB.

Unikali saugyklos variklių architektūra. MySQL siūlo keletą unikalių saugyklų variklių, kurie skirti specifiniams aplikacijų reikalavimams atitikti. Vienas iš jų – InnoDB – skirtas labai dideliems transakcijų kiekiams apdoroti. Kitas – MyISAM – saugyklos variklis skirtas palaikyti didelio apkrovimo interneto puslapius su terabaito dydžio duomenų saugyklomis. Kadangi MySQL yra atvirojo kodo, vartotojas pats gali kurti savo saugyklų variklius ir juos naudoti.

Stiprusis saugumo modelis. MySQL siūlo išskirtines saugumo galimybes:

- vartotojo autentifikavimas kliento aparatūrinės dalies lygyje;
- smulki vartotojo/objekto privilegijų sistema;
- keletas kodavimo ir atkodavimo metodų.

Spartinančioji atmintis. MySQL garsėja greitu atsaku net į ypač sudėtingas užklausas dėl aukštų atminties spartinimo charakteristikų. Keletas gerai konfigūruotų atminties spartintojų padeda reikalingą informaciją laikyti operatyvioje atmintyje. Tokiu būdu pasikartojančios užklaustos atliekamos akimirksniu. Konfigūruojant atminties spartintojus, nereikia sustabdyti/paleisti duomenų bazės serverio.

SQL standartas. MySQL naudoja platų SQL 99 standarto poaibį, taip pat daug praplėtimų.

VALDYMAS

Serveryje saugomos funkcijos. MySQL yra galimybė išsaugoti vartotojo sukurtas funkcijas ir jomis naudotis. Vienu metu jomis gali naudotis kelios aplikacijos. Taip išvengiama tinklo apkrovos, kadangi į serverį yra siunčiama viena užklausa vietoj kelių.

Trigeriai. MySQL trigeriai gali būti įgalinti prieš arba po to, kai DML (ang. Data Manipulation Language) atliko veiksmus su duomenų baze.

Šios savybės leidžia maksimaliai pritaikyti MySQL duomenų bazių serverį konkrečioms užduotims.

TRANSAKCIJOS

MySQL siūlo vieną iš galingiausių transakcijų duomenų bazių variklių. Įdiegtas InnoDB saugyklos variklis teikia įvairias galimybes, valdant aukšto lygio OLTP (ang. Online Transaction Processing) darbą.

ACID transakcijos. MySQL palaiko ACID (ang. Atomic Consistent Isolated Durable) transakcijų modelį[1]:

- Atomic (liet. atominis) – transakcijos atliekamos arba neatliekamos visapusiškai, t. y. jei įvyksta klaida, visas transakcijos padarytas darbas grąžinamas į pradinę būseną;
- Consistent (liet. konsistencija) – transakcija turi transformuoti sistemą iš vienos konsistencijos būsenos į kitą;
- Isolated (liet. izoliuotas) – kiekviena transakcija turi vykti nepriklausomai nuo kitų vykdomų transakcijų tuo pačiu metu;
- Durable (liet. patvarumas) – atlikta transakcija turi išlikti nepakitusi, net jei ir įvyko sistemos klaida.

Dviejų fazių patvirtinimo protokolas. Koordinatorius siunčia transakcijas keliems vykdytojams, kiekvienas iš jų atlieka veiklą ir siunčia koordinatoriui atsakymą, ar jam pavyko atlikti užduotį. Visiems vykdytojams atlikus darbą teisingai, koordinatorius siunčia patvirtinimą pakeitimams išsaugoti. Pakeitimai nesaugojami, jei nors vienam vykdytojui nepavyko atlikti darbo.

Mažų nesutarimų sprendimas. Eilutės lygio rakinimas yra InnoDB pakete kaip automatinis aklavietės detektorius. InnoDB taip pat siūlo multikonkurencijos kontroliavimą. Ši funkcija užtikrina tai, kad skaitytojas neblokuos rašytojo. Tai padidina duomenų prieinamumą.

Serveriu pagrįsta duomenų integracija. Klaidingi duomenys (klaidingos datos, skaičiai ir panašiai) gali būti automatiškai atmesti, naudojant stulpelio lygio taisyklių aprašus. Išorinių raktų tikrinimas užtikrina korektišką duomenų integraciją.

Aukšto lygio indeksavimas. InnoDB transakcijų variklis palaiko B-Tree indeksavimą taip pat puikiai kaip ir klasterių indeksus, kurie yra labai panašūs į Oracle indeksavimo variklį.

Automatinis saugyklos valdymas. MySQL saugyklos valdymo sistema automatiškai išplečia duomenų bazės saugyklą tiek, kiek reikia įrašomiems duomenims išsaugoti. InnoDB saugyklos variklis automatiškai paskirsto atmintį iš panaikintų objektų naujiems, taip eliminuojamas tuščių duomenų bazės puslapių skanavimas. InnoDB sugeba valdyti ypač dideles, iki 64 TB, duomenų bazines.[1]

Paskirstytosios transakcijos. MySQL visiškai palaiko XA transakcijas, aprėpdamas kelis MySQL serverius, tokios paskirstytos transakcijos nesunkiai palaikomos.

Aukštų charakteristikų struktūra. MySQL siūlo DML (duomenų manipuliavimo kalbą), kuri sumažina laiko tarpą, per kuri atliekamos transakcijos.

Duomenų kopijos ir jų atstatymas. Visoms MySQL duomenų bazėms gali būti daromos kopijos, kurios, esant būtinybei, būtų atstatytos. InnoDB siūlo specializuotą kopijos darymą, nesustabdant vykstančių darbų su duomenimis.

1.2.2. PostgreSQL – duomenų bazės serveris

PostgreSQL yra į objektus orientuotas duomenų bazės serveris, naudojantis lanksčią BSD stiliaus licenciją. Tai yra gera alternatyva nemokomoms sistemoms (tokioms kaip MySQL ar Firebird), taip pat ir garsioms mokamoms sistemoms (tokioms kaip Oracle, Sybase, Microsoft SQL).[2] PostgreSQL nėra kontroliuojamas vienos įmonės, tačiau priklauso plačiai kūrėjų/vartotojų visuomenei.

ARCHITEKTŪRA

PostgreSQL yra nepriklausomas nuo procesoriaus tipo, taip pat puikiai veikia įvairių operacinių sistemų aplinkose, įskaitant Windows, Unix, FreeBSD ir t. t.

Naudojant koduotąjį (SSL) prisijungimą prie duomenų bazės, užtikrinamas duomenų saugumas tiek siunčiant duomenys į serverį, tiek iš jo gaunant.

VALDYMAS

PostgreSQL turi galimybę programos blokus saugoti ir paleisti pačiame serveryje. Funkcijos gali būti rašomos PL/pgSQL, PL/Perl, C, C++ bei Java kalbomis.

Vartotojas gali kurti įvairius objektus, tokius kaip:

- Indeksai;
- Operatoriai (esami gali būti pakeisti vartotojo sukurtais);
- Domenai;
- Duomenų tipai, kurie gali būti indeksuojami per PostgreSQL Gist infrastruktūrą.

Trigeriai. Jie visiškai veiksniūs ir gali būti pritaikyti tiek lentelėms, tiek peržiūroms (view). Daugialypiai trigeriai yra pateikti abėcėlės tvarka. Papildomai kreipiantis, trigeriai gali paleisti funkcijas, kurios turėtų būti parašytos PL/Perl programavimo kalba.[2]

Spartinančioji atmintis. Spartinančiosios ir rūšiavimo atminčių dydžiai gali būti keičiami vartotojo. Taip vartotojas pritaiko PostgreSQL serverį konkrečiai duomenų bazei.

TRANSAKCIJOS

PostgreSQL siūlo ACID principą visoms transakcijoms. Įrašo lygio rakinimas ir mirties taško aptikimas, kurią turi PostgreSQL, užtikrina, kad sistema dirbtų teisingai.

Indeksai. Vartotojas gali kurti savo indeksus arba kurti juos, naudojantis B-Tree technologija. Skirtingi ir GiST indeksai taip pat galimi. PostgreSQL indeksai turi tokias savybes:

- Gebėjimas skenuoti/naudoti indeksus atbuline tvarka;
- Vykdomi indeksai – tai funkcijos ar procedūros rezultatas;
- Daliniai indeksai – indeksuoja tik dalį lentelės. Jie kuriami konkrečiai užklausiai;
- Indeksų masyvai – tai leidžia skaityti kelis indeksus, generuoti indeksų masyvą, kuris išreiškia jų persidengimo taškus, atitinkančius užklaustos kriterijus.

Dviejų fazių patvirtinimo protokolas. Tai papildomas transakcijų patikimumo modelis, kai transakciją atlieka keli vykdytojai. Sukurtą užduotį visi vykdytojai privalo atlikti vienodai, jei tai nepavyko nors vienam vykdytojui transakcija atmetama ir duomenys nėra iškraipomi.

Konkurencija. PostgreSQL valdo užklausų konkuravimą, pasikliaudamas MVCC (Multi-Version Concurrency Control). Tai kiekvienai užklausiai pateikia tuometinį duomenų bazės vaizdą, leidžia padaryti savo pakeitimus, būnant nematomam kitų užklausų atžvilgiu iki tol, kol transakcija vyksta.[2]

Duomenų kopijos ir jų atstatymas. PostgreSQL siūlo tris būdus duomenų bazės kopijai daryti:

- SQL dump;
- Failų lygio kopija;
- Kopija realiame laike.

KITOS SAVYBĖS

- Paveldimumas. Lentelės gali paveldėti charakteristikas iš „tėvo“ lentelės. Duomenys tarp „tėvo“ ir „vaiko/vaikų“ lentelių yra prieinami. Jei duomenys buvo koreguoti „vaiko“ lentelėje, „tėvo“ lentelėje bus atlikti tie patys veiksmai. Tai galioja viešiesiems su duomenimis susijusiems veiksams.

- Priklausomybių ryšiai – išoriniai raktai, stulpelių charakteristikos, įrašų kontrolė;
- Peržiūros (views) – užklaustos atsakymas patalpintas į lentelę;
- Visiški, vidiniai, išoriniai sujungimai;
- Sub užklaustos;
- Didelio dvejetainių ir teksto duomenų saugojimas;
- Domenai;
- Saugos taškai;
- Atstatymas laike;
- Naudoja aukšto lygio SQL 2003 standartą.

1.2.3. Firebird – duomenų bazės serveris

Firebird 1.5 – visiškai aprūpintas SQL duomenų bazės serveris. Tai galingas, tačiau nereikalaujanti didelių konfigūravimo ar administravimo darbų, duomenų bazės serveris. Firebird tinka tiek darbui vienam vartotojui su viena duomenų baze, tiek verslui. Vienas Firebird serveris gali palaikyti kelias duomenų bazes su galybe prisijungusių klientų.[3]

ARCHITEKTŪRA

OS priklausomybė. Dažniausiai Firebird naudojamas su Linux ir MS Windows operacinėmis sistemomis, įskaitant terminalinius serverius ir Citrix, taip pat Mac OS X, SOLARIS ir HP-UX gali dirbti su Firebird. Migracija iš vienos operacinės sistemos į kitą yra galima ir neskausminga, tereikia padaryti transportuojamą kopiją ir ją atstatyti jau kitoje operacinėje sistemoje.[3]

Fiziniai limitai. Firebird palaiko dideles duomenų bazes. Duomenų bazė gali būti padalinta į kelis failus, kiekvieno iš tokių failų dydis galimas iki 64 TB.

Serverio variklio versijos. Firebird turi tris skirtingas variklio versijas, visos jos gali būti pakeistos kitu varikliu ir kiekviena turi savo stipriąją pusę:

- Classic server. Viena instancija kliento prisijungimui, SMP/HT palaikomas;
- Superserver. Visi prisijungimai valdomi vieno modulio, SMP/HT nepalaikomas;
- Embedded versija. Visas variklis vienoje bibliotekoje, skirtas vienam vartotojui naudotis viena duomenų baze.

Visos versijos dirba tuo pačiu duomenų bazės formatu, tad bet kada galima pakeisti variklį kitu.

VALDYMAS

Praplėtimas. Naudojant Firebird'o PSQL (Procedural SQL), galima kurti galingas procedūras, kurios bus saugomos ir vykdomos serveryje. Išorinės funkcijos – tai vartotojų sukurtos funkcijos, UDF (User Defined Function) bibliotekos. Jos turėtų būti kuriamos C, C++ arba Delphi programavimo kalbomis. Sukurta biblioteka praplečia Firebird galimybes pagal konkrečią užduotį.

Trigeriai. Kiekviena lentelė gali turėti daug eilutės lygio trigerių, kurie vykdomi prieš arba po įterpimo, ištrynimo ar koregavimo. Trigerio viduje PSQL galimybės gali būti panaudotos užtikrinti duomenų korektiškumui, nustatyti reikšmes pagal nutylėjimą, įvairias išimtis ir daug kitų veiksmų. Universalūs trigeriai – tai vienas trigeris, kuris valdo tiek įterpimus, tiek koregavimus ar trynimus vienu metu.

TRANSAKCIJOS

Transakcijų valdymas. Viena kliento aplikacija gali turėti keletą konkuruojančių transakcijų. Firebird siūlo kliento valdomus izoliacijos lygius. Dviejų fazių patvirtinimo

protokolas darbe su duomenų baze užtikrina nuoseklumą. Optimistinis užrakinimas palaikomas taip pat, kaip ir kelių transakcijų atstatomieji taškai.[3]

ACID transakcijos. Firebird palaiko ACID transakcijų modelį. Jis nagrinėtas, aprašant MySQL duomenų bazių serverį.

Dviejų fazių patvirtinimas. Firebird siūlo duomenų korektiškumo protokolą, kur transakciją atlieka keli vykdytojai. Jų veiklos rezultatai lyginami, jei jie nesutampa, transakcija priimama kaip klaidinga ir pakeitimai duomenų bazėje nevykdomi.

Konkurencija. MGA – tai speciali Firebird funkcija, kuri užtikrina, kad variklis valdytų kelias tos pačios fizinės atminties versijas. Taigi kiekviena transakcija turi savo duomenų versiją, nepriklausomą nuo kitų transakcijų (skaitytojas nekliudys rašytojo darbui). Tai užtikrina OLTP (ang. Online Transaction Processing) bei OLAP (ang. On Line Analytical Processing).

Duomenų kopijos ir jų atstatymas. Realus laiko dubleriai – nėra būtina sustabdyti duomenų bazės serverį kopijai padaryti. Dubliavimo procesas sukuria momentinį duomenų bazės būklės vaizdą. Taigi klientai gali toliau dirbti su duomenų baze, kol tuo metu daroma kopija.

KITOS SAVYBĖS

- Prisijungimas. Firebird turi daug prisijungimo variantų, įskaitant į C/C++ ar Delphi integruotus paketus, ODBC, JDBC, PHP, dbExpress, .NET ir per tiesiogines API užklausas.
- Visiškas SQL 99 standarto palaikymas. Į Firebird įterpta daugelis SQL2003 standarto funkcijų.
- Licenzija. Naudojama IPL (ang. InterBase Public License) licenzija, kuri yra visiškai nemokama naudojant tiek asmeniniams, tiek komerciniams tikslams.

1.2.4. Atvirojo kodo duomenų bazių serverių panašumai ir skirtumai

Aptarti trys duomenų bazės serveriai – MySQL, PostgreSQL ir Firebird. Visos sistemos yra pirmaujančios tarp atvirojo kodo duomenų bazių serverių. Jų savybės yra labai panašios, skiriasi tik kelios funkcijos arba jos vadinamos skirtingais vardais, nors atlieka tą pačią funkciją.

Užklausų spartinančios atminties valdymą turi ir MySQL, ir PostgreSQL, tačiau Firebird neturi galimybės ją valdyti. Visos trys sistemos palaiko ACID standartą, transakcijas, taip pat turi savitus konkurencijos valdymo paketus. Indeksus palaiko visos trys sistemos, tačiau apie B-Tree indeksavimą Firebird neužsimena. Tokios funkcijos, kaip duomenų kopijos darymas, nestabdant darbo, praplėtimas vartotojo kurtomis funkcijomis, laikinos lentelės (views) ar dviejų fazių darbo patvirtinimas yra visų trijų sistemų dokumentacijose.

2 lentelė. Duomenų bazių serverių palyginimas

	MySQL	PostgreSQL	Firebird
Spartinančiosios atminties valdymas	TAIP	TAIP	NĖRA
Transakcijos	TAIP	TAIP	TAIP
ACID	TAIP	TAIP	TAIP
Duomenų kontrolė	TAIP	TAIP	TAIP
Maksimalus duomenų bazės dydis	64TB	NENURODYTA	64TB
B-TREE indeksavimas	TAIP	TAIP	NĖRA
Paskirstytos duomenų bazės	TAIP	TAIP	TAIP
Kopijos darymas, nestabdant darbo	TAIP	TAIP	TAIP
Praplėtimas	TAIP	TAIP	TAIP
Trigeriai	TAIP	TAIP	TAIP
Dviejų fazių darbo patvirtinimas	TAIP	TAIP	TAIP
Laikinos lentelės (views)	TAIP	TAIP	TAIP
Užklausų konkurencijos valdymas	YRA	MVCC	MGA
SQL standartas	SQL99	SQL 2003	SQL 2003

Šaltinis: sukurta autoriaus.

Transakcijos. Tokios ypatybės kaip ACID standartas, dviejų fazių patvirtinimo protokolas, konkurencijos valdymas ar duomenų kopijos ir atstatymas yra palaikomos visų analizuotų duomenų bazių serverių.

Valdymas. Visi analizuoti duomenų bazių serveriai turi galimybę saugoti ir paleisti vartotojo kurtas funkcijas, taip pat kaip standartas palaikomi trigeriai. MySQL ir PostgreSQL turi galimybę valdyti spartinančią atmintį.

Architektūra. Visi analizuoti duomenų bazių serveriai gali būti įdiegti įvairaus tipo sistemose. MySQL ir Firebird turi po keletą serverio variantų, o PostgreSQL turi tik vieną variantą.

Vienintelis MySQL palaiko SQL99 standartą, kai PostgreSQL ir Firebird palaiko SQL 2003 standartą.

1.3. Vertinimo kriterijai

Atlikus panašių sistemų analizę galima teigti, kad pagrindiniai sistemos vertinimo kriterijai yra vartotojo sąsaja, vartotojo patogumas bei sistemos greitis. Aptartos panašios sistemos parodė pagrindinius aspektus į ką reikia atkreipti dėmesį:

- Prekių sąrašo grupavimo pagal prekės būseną;
- Pastovių prekės būsenų naudojimas;
- Gedimo registravime naudojamas laukų užpildymą pagal serijos numerį;
- Atsidarius gedimą matyti, kada prekė perėjo į kiekvieną būseną;
- Darbų akto spausdinimas;
- Vartotojo sąsaja.

Kiekvienai sistemai svarbus greitis, daugumai apskaitos sistemų daugiausiai laiko sugaištama duomenims pasiekti, bei pakeitimams saugoti. Svarbu pasirinkti tinkamą duomenų bazės serverį.

Tiriami trys atviro kodo duomenų bazių serveriai. Tam duomenų bazės serveriais reikia atlikti įvairias operacijas, stebėti atlikimo greitį, gebėjimą išnaudoti esamus resursus. Bandymai atliekami su jau sukurta aptarnavimo paslaugų valdymo sistema. Parinkti trys atviro kodo duomenų bazių serveriai: MySQL, PostgreSQL ir Firebird. Šios sistemos testuojamos pagal tam tikrus kriterijus.

Pasirinkti testai:

- Duomenų įrašymo greitis;
- Duomenų bazės dydžio kitimas didėjant įrašų skaičiui;
- Paieška nurodant visą lauką;
- Paieška nurodant fragmentą;
- Paieška ir koregavimas (pagal ID);
- Paieška ir koregavimas (pagal visą lauką);
- Paieška ir koregavimas (pagal fragmentą);
- Trynimas (pagal visą lauką fragmentą);
- Trynimas (pagal);

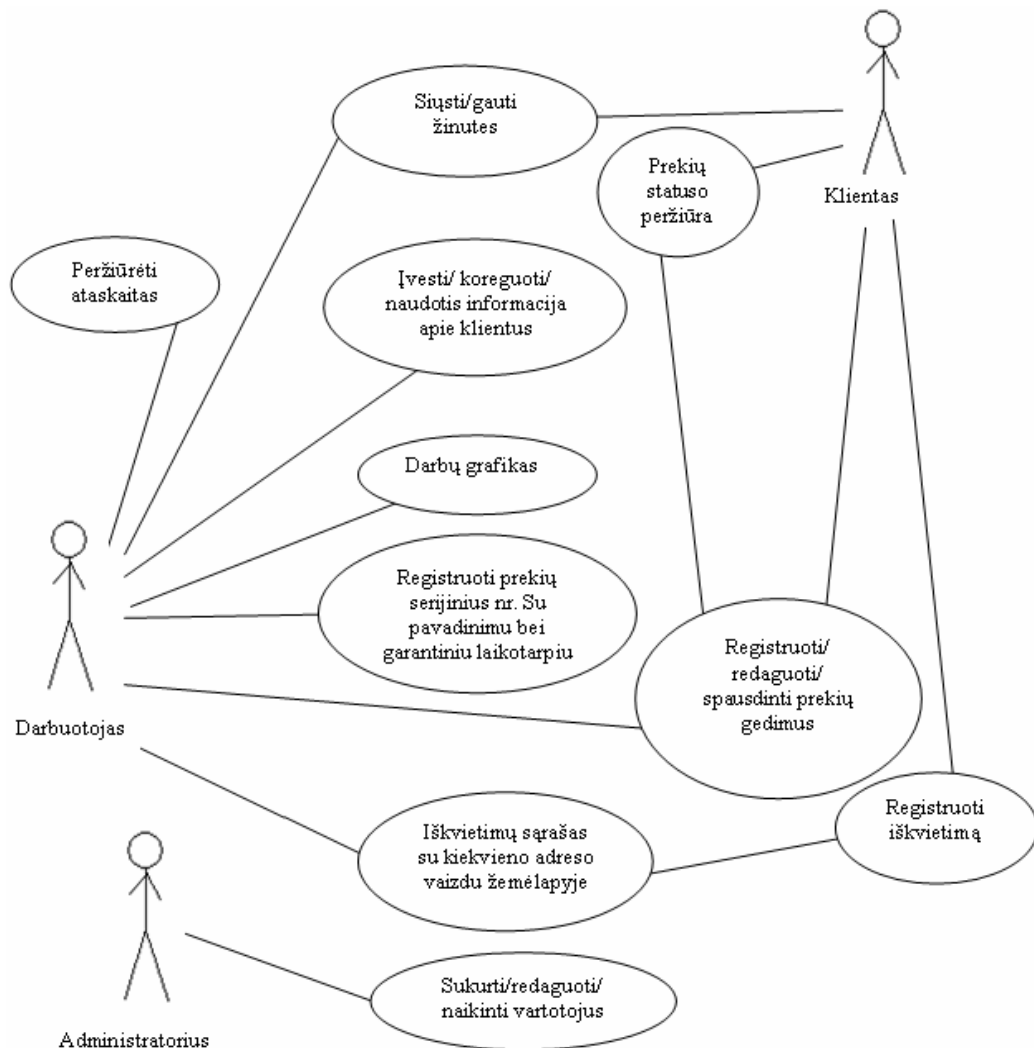
Gauti rezultatai susisteminti ir pateikti, naudojant įvairius grafikus bei grafines priemones.

2. Aptarnavimo paslaugų teikimo valdymo sistemos projekto modelis

Aptarnavimo paslaugų teikimo valdymo sistema yra web tipo sistema. Sistemos vartotojo sąsaja realizuota, naudojant HTML, veiksmai su duomenimis realizuoti PHP programavimo kalba, web serveris – Apache, duomenų saugojimui pasirinkta MySQL duomenų bazė. HTML pasirinktas, nes sistema yra prieinama internetu.[10] PHP programavimo kalba pasirinkta dėl atvirojo kodo licenzijos[12, 13]. Duomenų bazės serveris buvo pasirinktas iš trijų populiariausių atvirojo kodo duomenų bazių serverių, t. y. MySQL, PostgreSQL ir Firebird. Atlikus tyrimus, bus nustatyta, ar MySQL duomenų bazės serveris labiau tinka aptarnavimo paslaugų teikimo valdymo sistemai nei PostgreSQL ir Firebird.

2.1 Sistemos panaudos atvejai

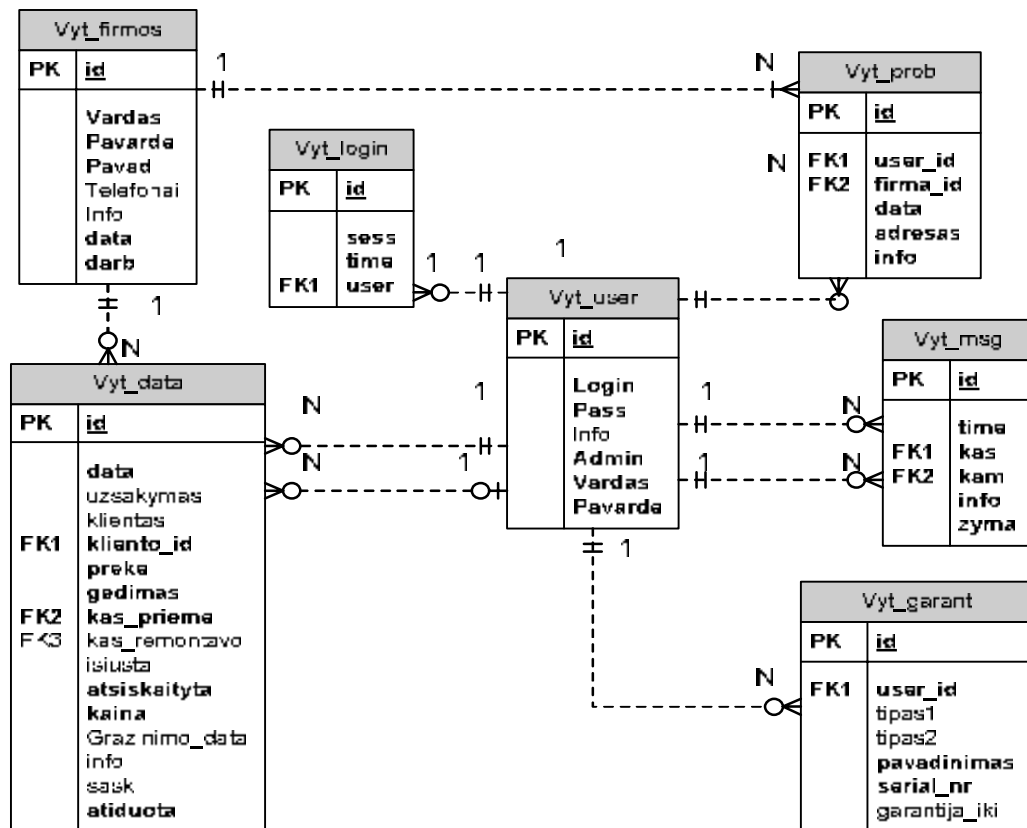
Ribas tarp sistemos ir vartotojo nusako panaudojimo atvejų diagrama. Panaudojimo atvejų diagrama sudaryta, įvertinant kiekvieną išskirtą veiklos įvykį ir kuriamos sistemos indėlių į šį įvykį.



1 pav. Panaudos atvejų diagrama

2.2 Duomenų bazė

Kaip matoma paveiksle nr.2, aptarnavimo paslaugų teikimo valdymo sistema naudoja septynias duomenų bazės lenteles, kurios yra tarpusavyje susijusios.



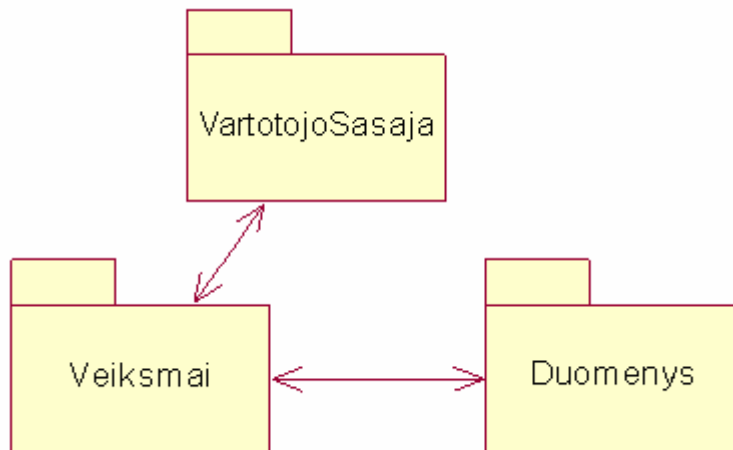
2 pav. Duomenų bazės esybių ryšių diagrama

Trumpas esybių aprašymas:

- **Vyt_firmos** – esybė, atsakinga už klientų informacijos saugojimą;
- **Vyt_data** – esybė, atsakinga už gedimų informacijos saugojimą;
- **Vyt_login** – esybė, atsakinga už prisijungimo informacijos saugojimą;
- **Vyt_user** – esybė, atsakinga už vartotojų informacijos saugojimą;
- **Vyt_prob** – esybė, atsakinga už iškvietimų informacijos saugojimą;
- **Vyt_msg** – esybė, atsakinga už žinučių informacijos saugojimą;
- **Vyt_garant** – esybė, atsakinga už naujų prekių informacijos saugojimą.

2.3 Sistemos loginė architektūra

Sistema realizuota, naudojant tris įrankius (HTML, PHP, MySQL), kiekvienam jų yra priskirta klasė ar jų grupė. Sistemos paketų diagrama pateikta paveiksle nr. 3.



3 pav. Sistemos loginė architektūra

Trumpas paketų aprašymas:

- **Vartotojo sąsaja** paketas turi klasę, atsakingą už sąsają su vartotoju;
- **Duomenys** paketo klasėse saugomi programai reikalingi duomenys;
- **Veiksmi** paketo klasės atsakingos už informacijos valdymą.

2.3.1 Paketo „Vartotojo Sąsaja“ klasė ir metodai

Vartotojo sąsajos klasė apibrėžia sistemos ir vartotojo bendradarbiavimą. Ši klasė turi aštuonis metodus, kiekvienas naudojamas, vartotojui atliekant atitinkamus veiksmus sistemoje.

Sasaja
+Vartotojai()
+Prisijungimas()
+Klientai()
+Nauja_preke()
+Gedimai()
+Iskvietimai()
+Ataskaitos()
+Zinutes()

4 pav. Vartotojo sąsajos klasė.

Vartotojo sąsajos metodai:

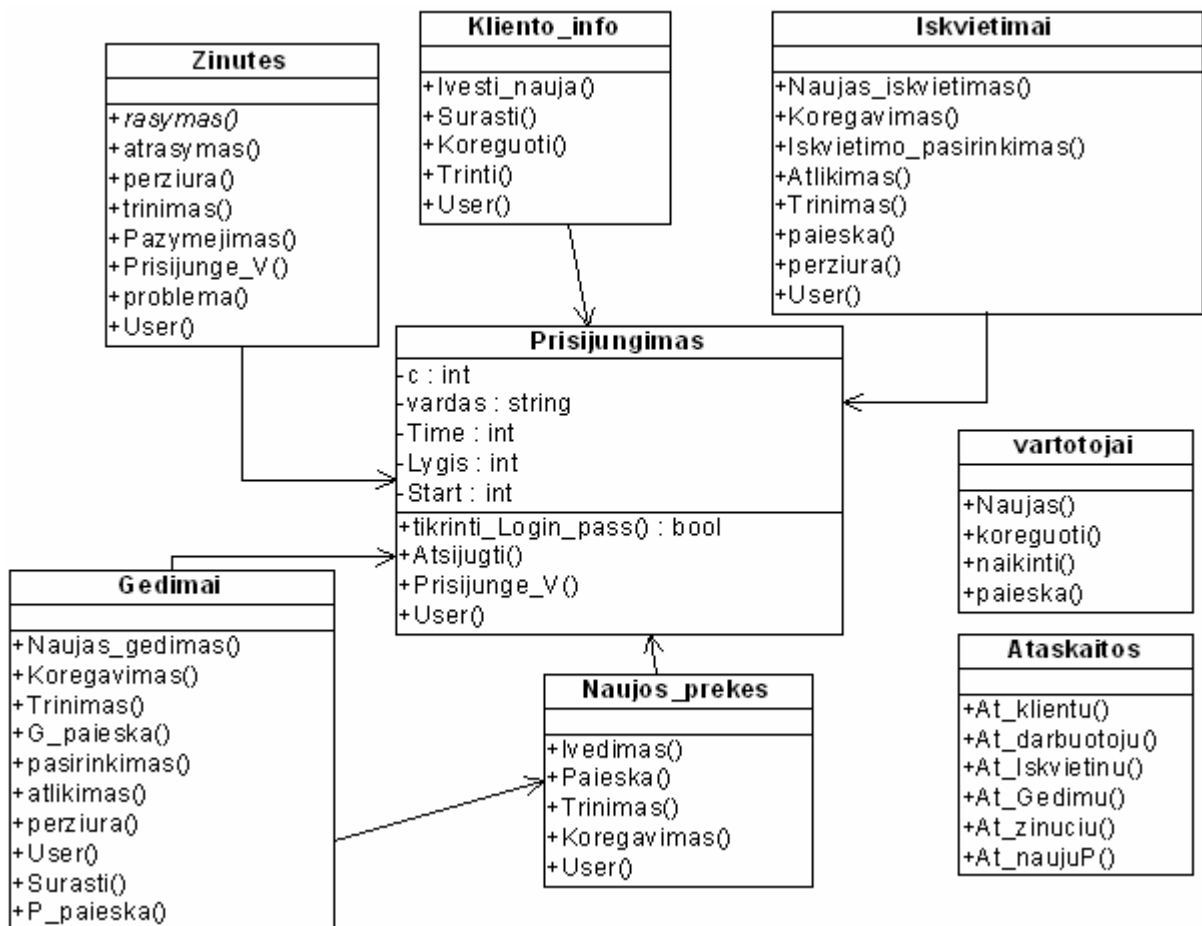
- **Vartotojai()** – naudojamas, administruojant informaciją apie sistemos vartotojus;
- **Prisijungimas()** – naudojamas, vartotojui jungiantis prie sistemos;
- **Klientai()** – naudojamas, administruojant informaciją apie sistemos klientus;
- **Nauja_preke()** – naudojamas, administruojant naujas prekes, t. y. registruojant, koreguojant;
- **Gedimai()** – naudojamas, administruojant su gedimais susijusią informaciją;

- **Iškvietimai()** – naudojamas, administruojant su iškvietimais susijusią informaciją;
- **Ataskaitos()** – naudojamas, formuojant bei vaizduojant įvairias ataskaitas;
- **Zinutes()** – naudojamas, žinučių siuntimui, gavimui bei kitoms operacijos su žinutėmis.

Kiekvienas šios klasės metodas, kreipiasi į atitinkamą „Veiksmai“ klasės metodą, kuris atitinkamai atlieka veiksmus su duomenimis, kuriuos saugo paketas „Duomenys“. Taigi kiekvienas veiksmas reikalauja greito duomenų bazės darbo.

2.3.2 Paketo „Veiksmai“ klasės ir metodai

Paketas „Veiksmai“ turi 8 klases, kurios vienaip ar kitaip bendradarbiauja tarpusavyje, taip apibrėždamos sistemos funkcionalumą.



5 pav. Paketo „Veiksmai“ klasių diagrama

Trumpas kiekvienos klasės aprašymas:

- Žinutės – klasė atsakinga už operacijas, susijusias su žinutėmis;
- Kliento_info – klasė, kurioje tvarkomi su klientu susiję duomenys;
- Iškvietimai – klasė, kuri tvarko iškvietimų srautus;
- Naujos prekės – klasė, kurioje tvarkoma informacija apie naujas prekes;
- Vartotojai – klasė, atsakinga už vartotojų administravimą;

- Gedimai – klasė, kurioje tvarkomas gedimų dokumentavimas;
- Ataskaitos – klasė, kuri formuoja ataskaitas;
- Prisijungimas – klasė, kuri administruoja vartotojų prisijungimą prie sistemos.

Klasės suskirstytos pagal panaudos atvejus, taip buvo išvengta painių klasių ryšių, sistema tapo aiškesnė ir lengviau suprantama. Kiekvienos klasės metodas kreipiasi į duomenų bazės serverį su atitinkama užklausa, tai gali būti įterpimo, trynimo, paieškos ar redagavimo užklausa. Metodų yra pakankamai daug ir visi jie reikalauja greito duomenų bazės serverio atsako, o aprašomas vienas duomenų bazės serveris.

2.3.3 Paketo „Tyrimas“ klasė ir metodas

Paketas „Tyrimas“ sukurtas specialiai magistro darbui atlikti, klasė pateikta paveiksle nr. 6. Ši klasė turi 25 metodus, vienas valdantis metodas ir kiti 24 metodai skirti atlikti pakeitimams duomenų bazėje.

Trumpas metodų aprašymas:

- Valdymas – Valdo visus likusius metodus;
- MySQL_iterp – vykdo įterpimą į duomenų bazę, valdoma MySQL DBVS;
- Firebird_iterp – vykdo įterpimą į duomenų bazę, valdoma Firebird DBVS;
- PostgreSQL_iterp – vykdo įterpimą į duomenų bazę, valdoma PostgreSQL DBVS;
- MySQL_koreg_ID – vykdo duomenų bazės įrašų koregavimą, nurodant ID, valdoma MySQL DBVS;
- Firebird_koreg_ID – vykdo duomenų bazės įrašų koregavimą, nurodant ID, valdoma Firebird DBVS;
- PostgreSQL_koreg_ID – vykdo duomenų bazės įrašų koregavimą, nurodant ID, valdoma PostgreSQL DBVS;
- MySQL_koreg – vykdo duomenų bazės koregavimą, nurodant visą lauką, valdoma MySQL DBVS;
- Firebird_koreg – vykdo duomenų bazės koregavimą, nurodant visą lauką, valdoma Firebird DBVS;
- PostgreSQL_koreg – vykdo duomenų bazės koregavimą, nurodant visą lauką, valdoma PostgreSQL DBVS;
- MySQL_koreg_kontekst – vykdo duomenų bazės koregavimą, nurodant lauko, kontekstą valdoma MySQL DBVS;
- Firebird_koreg_kontekst – vykdo duomenų bazės koregavimą, nurodant lauko, kontekstą, valdoma Firebird DBVS;

- PostgreSQL_koreg_kontekst – vykdo duomenų bazės koregavimą, nurodant lauko kontekstą, valdoma PostgreSQL DBVS;
- MySQL_paiseska – vykdo paiešką duomenų bazėje, nurodant visą lauką, valdoma MySQL DBVS;
- Firebird_paiseska – vykdo paiešką duomenų bazėje, nurodant visą lauką, valdoma Firebird DBVS;
- PostgreSQL_paiseska – vykdo paiešką duomenų bazėje, nurodant visą lauką, valdoma PostgreSQL DBVS;
- MySQL_paiseska_kontekst – vykdo paiešką duomenų bazėje, nurodant lauko kontekstą, valdoma MySQL DBVS;
- Firebird_paiseska_kontekst – vykdo paiešką duomenų bazėje, nurodant lauko kontekstą, valdoma Firebird DBVS;
- PostgreSQL_paiseska_kotetekst – vykdo paiešką duomenų bazėje, nurodant lauko kontekstą, valdoma PostgreSQL DBVS;
- MySQL_trynimas – vykdo trynima duomenų bazėje, nurodant visą lauką, valdoma MySQL DBVS;
- Firebird_trynimas – vykdo trynima duomenų bazėje, nurodant visą lauką, valdoma Firebird DBVS;
- PostgreSQL_trynimas – vykdo trynima duomenų bazėje, nurodant visą lauką, valdoma PostgreSQL DBVS;
- MySQL_trynimas_kontekst – vykdo trynima duomenų bazėje, nurodant lauko kontekstą, valdoma MySQL DBVS;
- Firebird_trynimas_kontekst – vykdo trynima duomenų bazėje, nurodant lauko kontekstą, valdoma Firebird DBVS;
- PostgreSQL_trynimas_kontekst – vykdo trynima duomenų bazėje, nurodant lauko kontekstą, valdoma PostgreSQL DBVS.



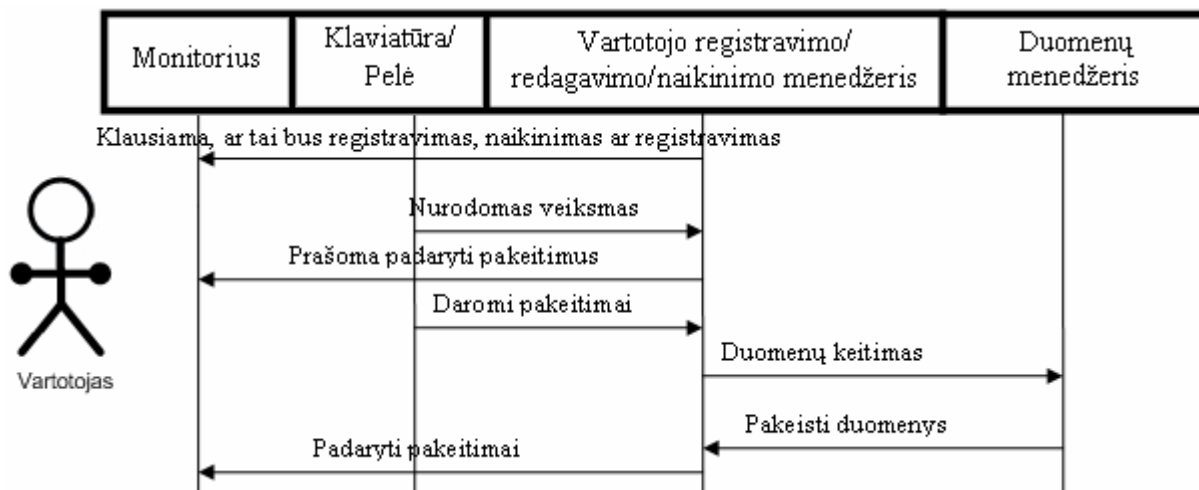
6 pav. Klasė „Tyrimas“ ir jos metodai

2.4 Dinaminis sistemos vaizdas

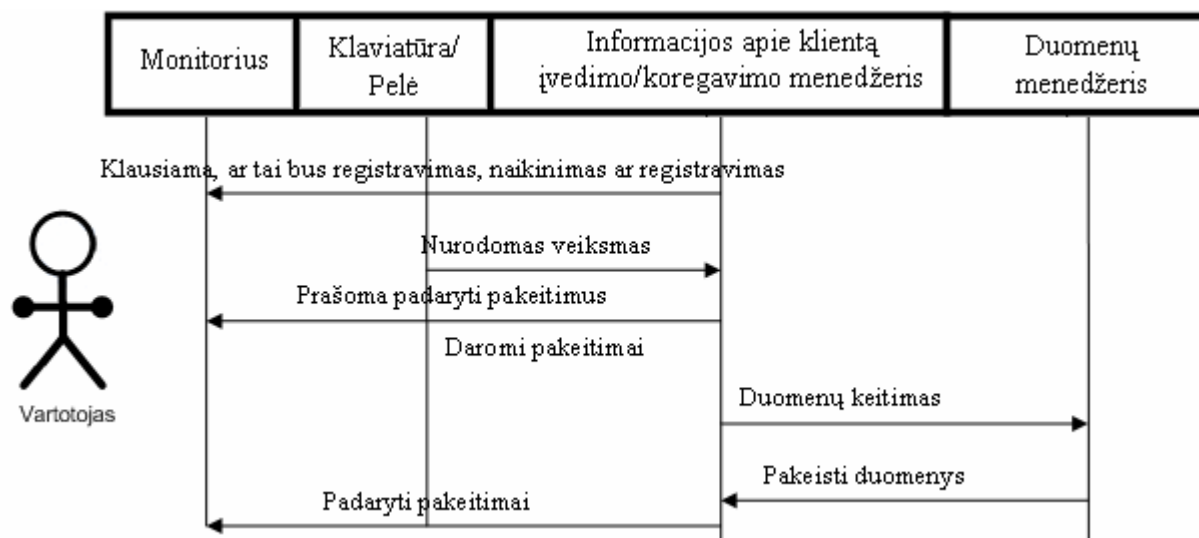
Pateikiamos sąveikos (interaction), būsenų (state) ir veiklos (activity) diagramos. Sąveikai pavaizduoti pakanka pasirinkti vieną iš dviejų diagramų – sekų (sequence) arba bendradarbiavimo (collaboration), tačiau esminiams sprendiniams pagrįsti galima pateikti ir abi. Pagrindinėmis pasirinkus sekų (bendradarbiavimo) diagramas, dokumentacijoje privalo būti bent viena bendradarbiavimo (sekų) diagrama.

2.4.1 Veiksmų eiliškumo diagramos panaudos atvejams

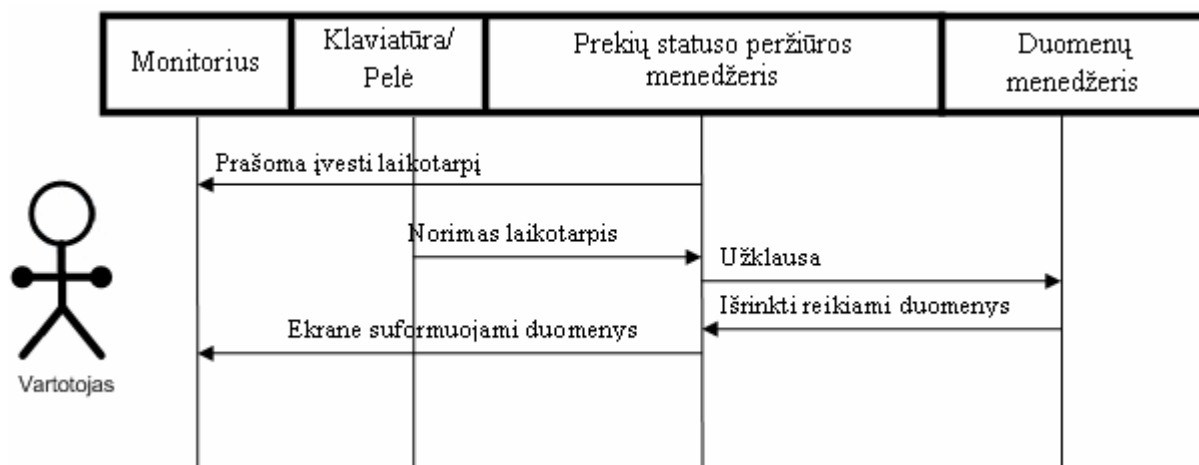
Kiekvienas panaudos atvejis turi savo veiksmų eiliškumo diagramą, kuri rodo, koku eiliškumu atliekama atitinkama operacija. Paveikslai nr 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 vaizduoja veiksmų eiliškumo diagramas kiekvienam panaudos atvejui.



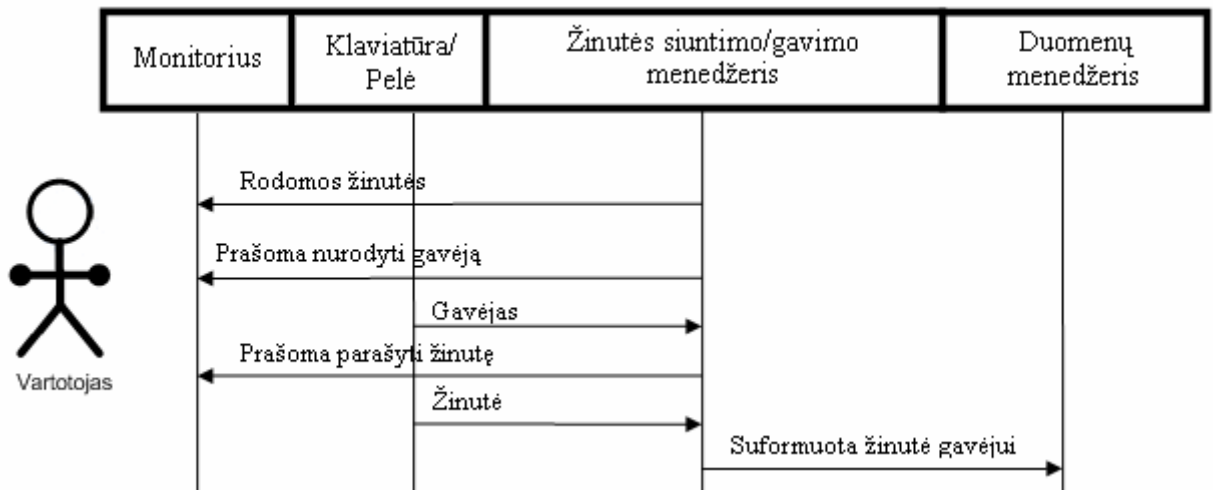
7 pav. Veiksmų eiliškumo diagrama vartotojų sukūrimui/redagavimui/naikinimui



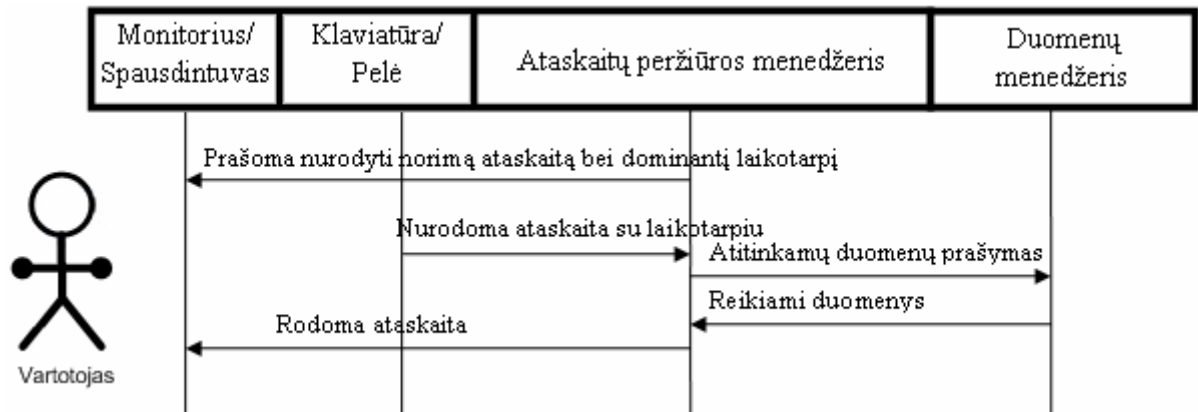
8 pav. Veiksmų eiliškumo diagrama informacijai apie klientą administruoti



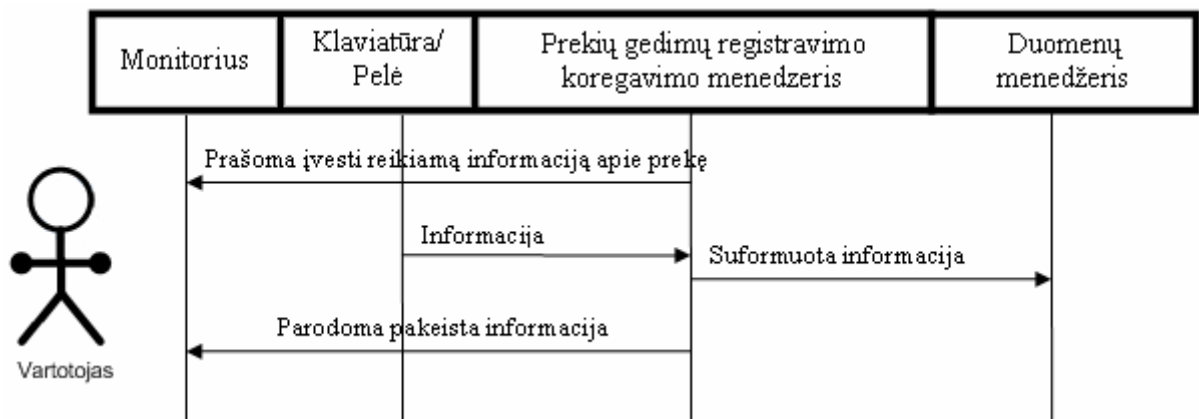
9 pav. Veiksmų eiliškumo diagrama prekių statusui peržiūrėti



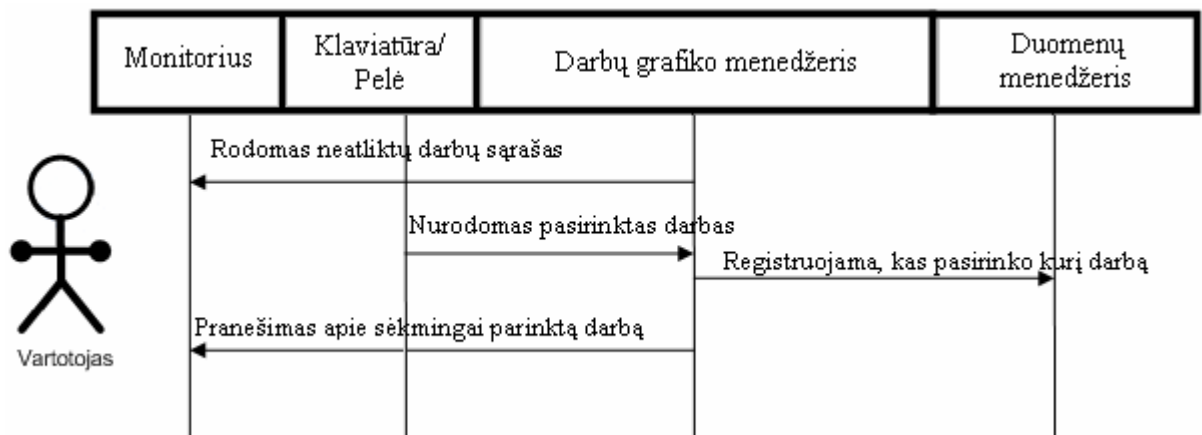
10 pav. Veiksmų eiliškumo diagrama žinutėms siusti/gauti



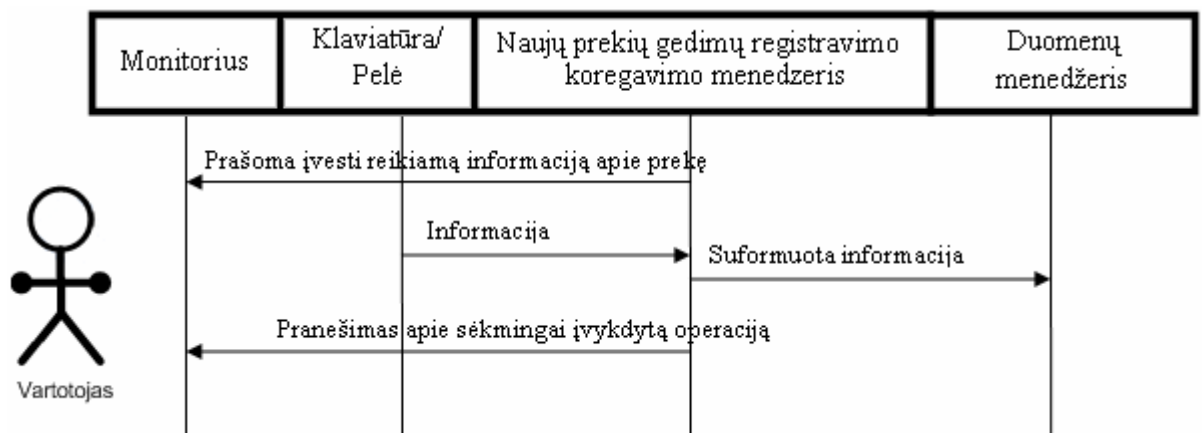
11 pav. Veiksmų eiliškumo diagrama ataskaitoms peržiūrėti



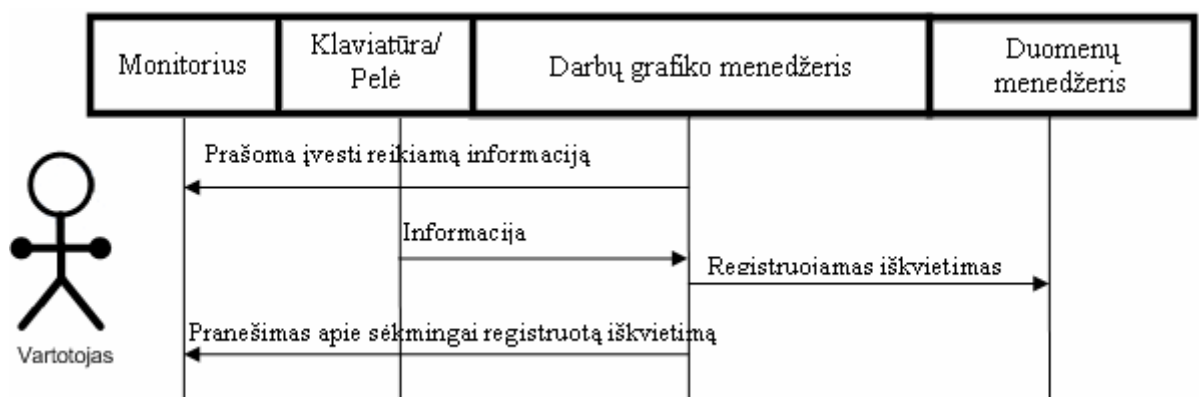
12 pav. Veiksmų eiliškumo diagrama prekių gedimui administruoti



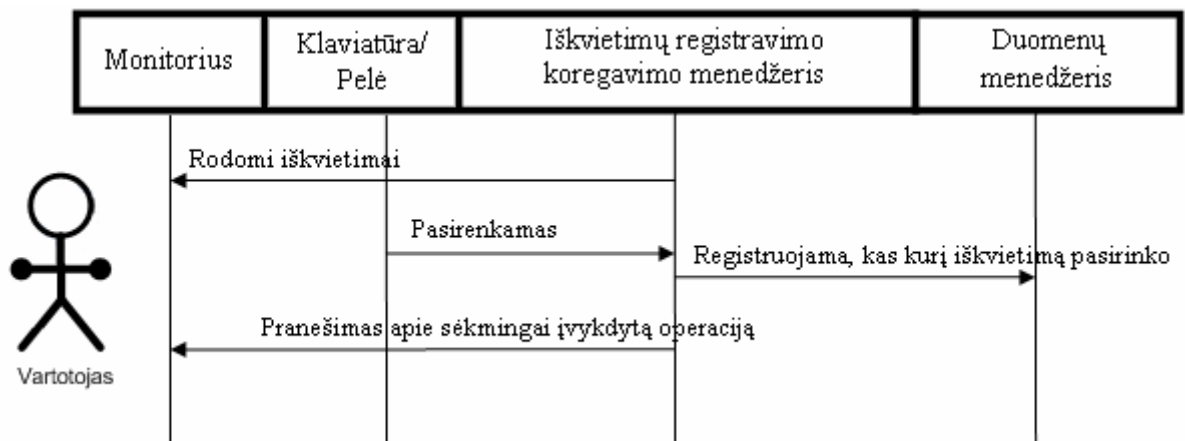
13 pav. Veiksmų eiliškumo diagrama darbų grafikai



14 pav. Veiksmų eiliškumo diagrama naujoms prekėms administruoti



15 pav. Veiksmų eiliškumo diagrama iškvietimą registruoti



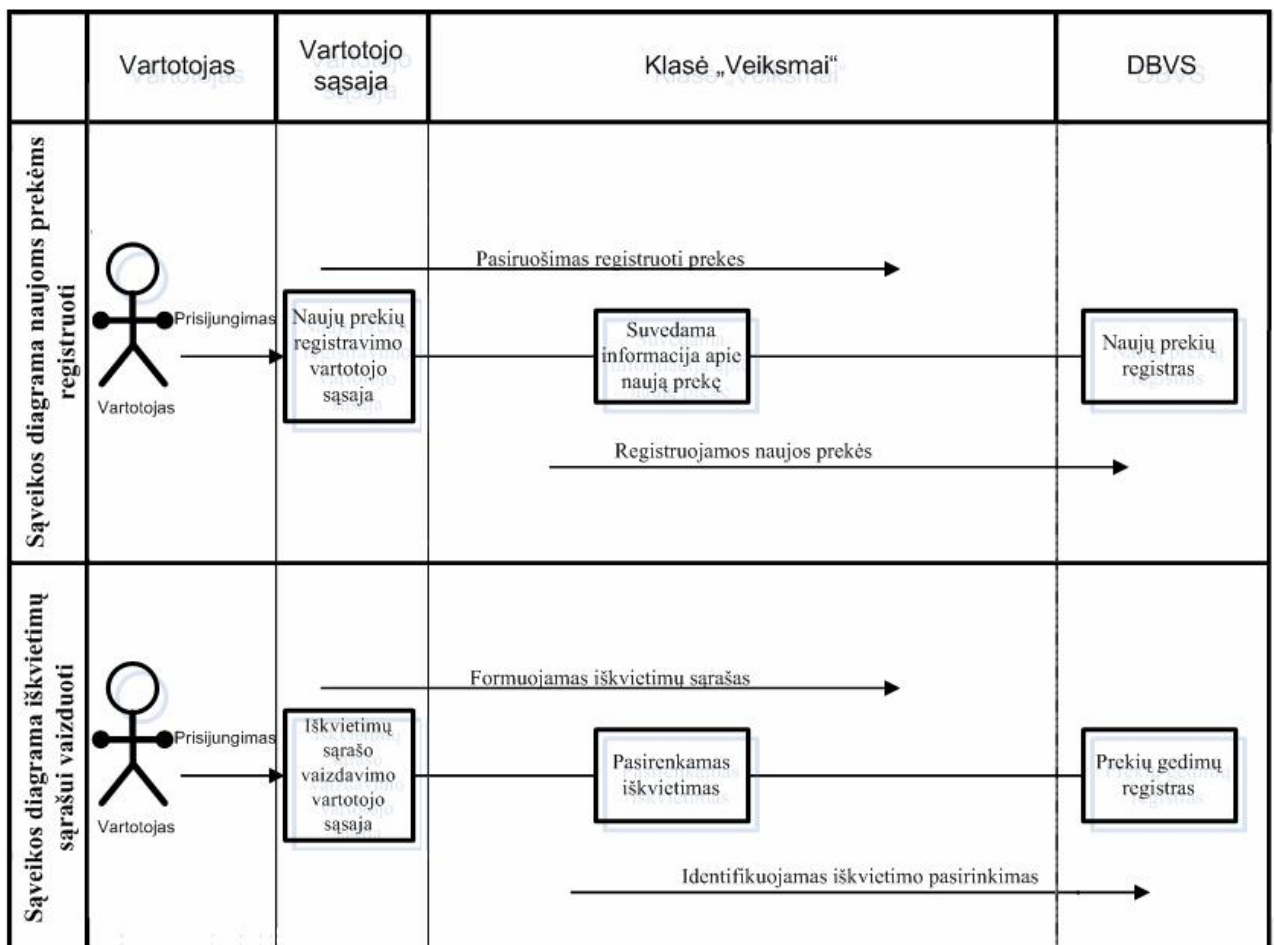
16 pav. Veiksmų eiliškumo diagrama iškviėtimų sąrašui su adresu žemėlapyje

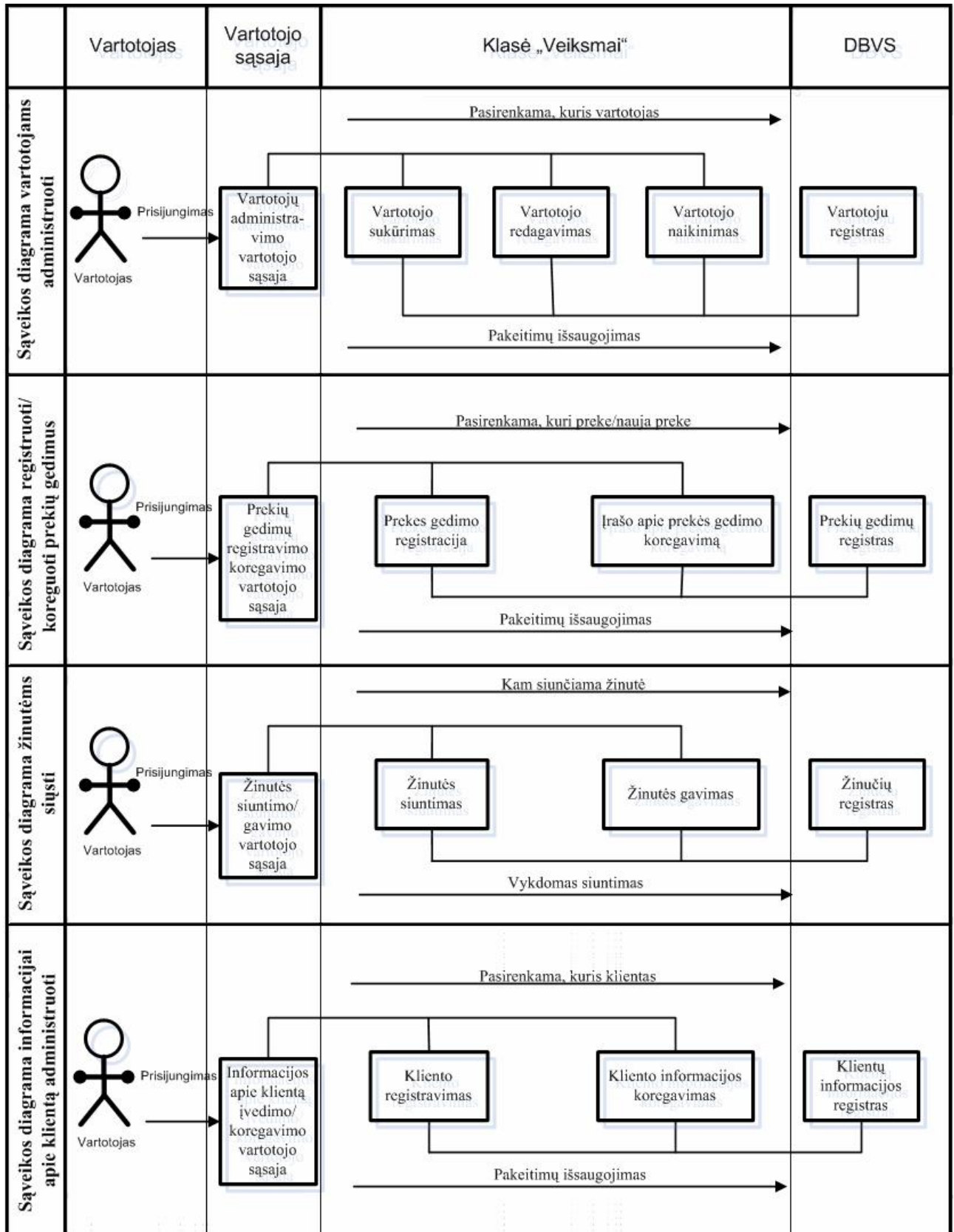
Iš aukščiau pateiktų veiksmų eiliškumo diagramų matome, kad kiekvienas panaudos atvejis naudoja duomenų menėdžerio paslaugas, t.y. reikalingas greitai dirbantis duomenų bazės serveris.

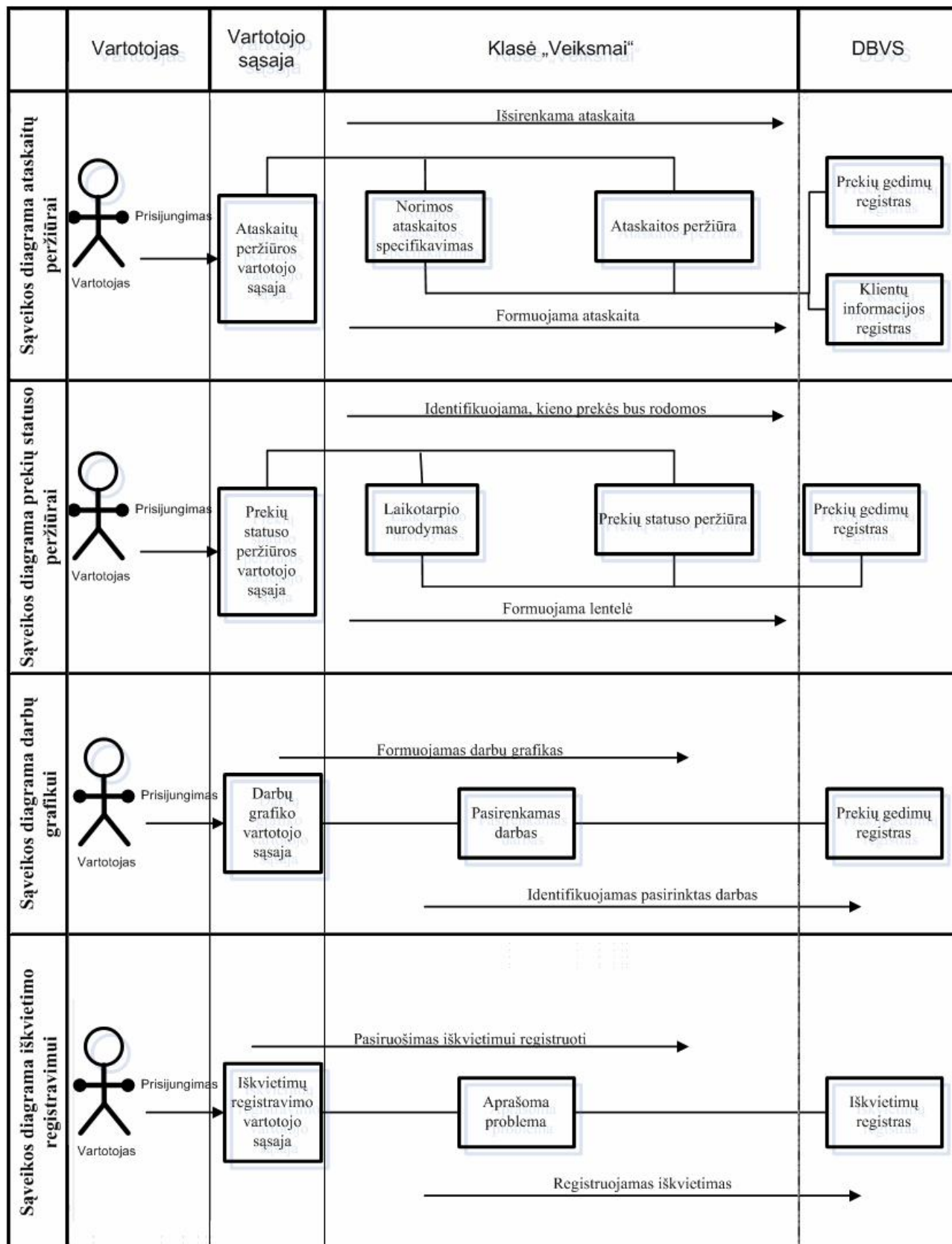
2.4.2 Sąveikos diagramos

Sąveikos diagramos parodo, kaip klasės sąveikauja tarpusavyje, kokie metodai taikomi norint atlikti vieną ar kitą operaciją. Visos sąveikos diagramos pateiktos lentelėje nr. 3.

3 lentelė. Sąveikos diagramos







3. Atvirojo kodo duomenų bazių serverių tyrimo rezultatai

Pagrindinis Tiriamojo darbo tikslas – išanalizuoti atvirojo kodo duomenų bazių serverius, labiausiai tinkančius aptarnavimo paslaugų teikimo valdymo sistemai. Pasirinkti populiariausi atvirojo kodo duomenų bazių serveriai: Firebird, MySQL ir PostgreSQL sistemos.

Tyrimo metu išanalizuotos įterpimo, koregavimo, koregavimo pagal kontekstą, paieškos, paieškos pagal kontekstą, trynimo bei trynimo pagal kontekstą funkcijos. Pagrindinis vertinimo kriterijus – darbo atlikimo greitis, taip pat atsižvelgiama į duomenų bazės dydžio priklausomybę nuo įterptų įrašų skaičiaus bei resursu išnaudojimą. Tyrimai atliekami vienu kompiuteriu, turinčiu Intel Centrino 1,70 Ghz procesorių, 512 MB operatyviosios atminties, kietąjį diską su 60 Gb pastovios atminties ir 5400 rpm greičio sukiais bei Windows Xp Professional operacinę sistemą.

Testai buvo paleidžiami iš aptarnavimo paslaugų teikimo valdymo sistemos, sukurtos specialaus tyrimams modulio. Pasibaigus kiekvienam testui, rezultatai buvo kaupiami ir apdorojami, vėliau pateikti grafiškai.

3.1. Įterpimo funkcijos tyrimas

Įterpimo funkcijos tyrimui buvo sukurti keli testai: 50 000 įrašų įterpimas, paliekant prieš tai įrašytus duomenis, kartotas 18 kartų, 100 000 įrašų įterpimas, paliekant prieš tai įrašytus duomenis, kartotas 9 kartus, įterpimas, paduodant kaskart didesnę įrašų kiekį, kartotas 5 kartus. Visų testų metu įterpimas buvo vykdomas į dvi lenteles, kiekvieno testo kartojimo metu įterpiami identiški įrašai visiems testuojamiems duomenų bazės serveriams.

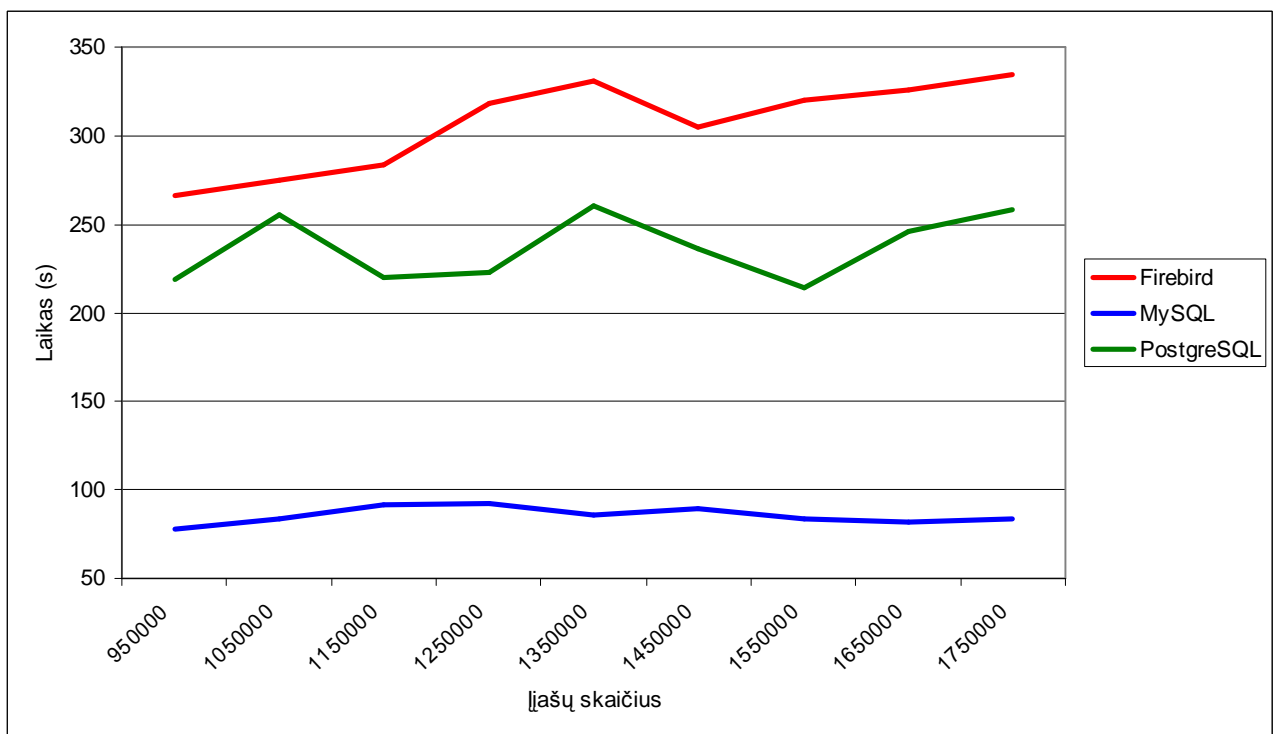
Atlikus tyrimą įterpiant po 50 tūkst. eilučių, pastebėta, kad nėra priklausomybės tarp duomenų bazės dydžio ir įterpimo greičio, t. y. nesvarbu, kokio dydžio yra duomenų bazė, įterpimo laikas išlieka apytikriai konstanta, kas matoma. paveiksle nr. 17.

Atliktas tyrimas parodė, kad greičiausiai įterpimo veiksmą po 50 tūkst. atlieka MySQL duomenų bazės serveris, jis vidutiniškai sugaišo apie 38 sekundes ir tai atliko greičiau nei per minutę, kas yra tris kartus greičiau nei kiti analizuojami duomenų bazių serveriai. Firebird ir PostgreSQL duomenų bazės serveriai įterpimą po 50 tūkst. atliko atitinkamai per 123 sekundes ir 118 sekundes (iš 18 testų paimtas vidurkis).



17 pav. Įterpimas po 50 tūkst. eilučių

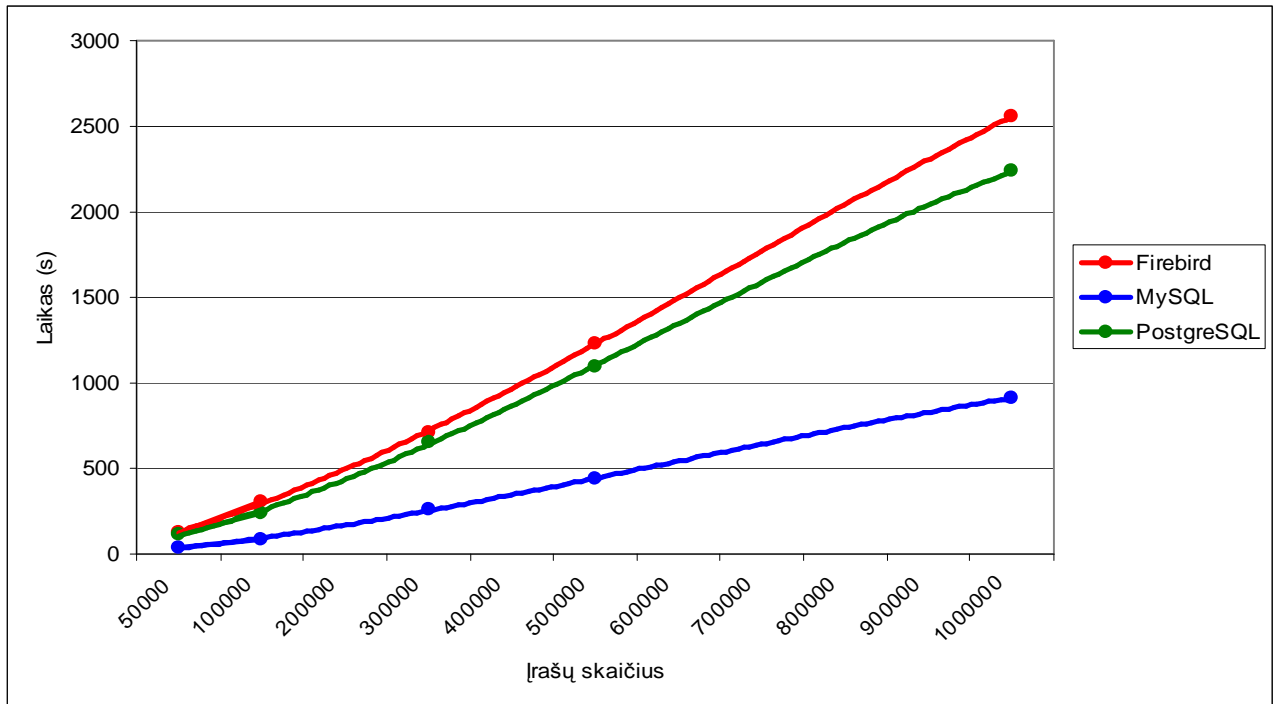
Atlikus tyrimą įterpiant po 100 tūkst. eilučių, pastebėta, kad priklausomybė tarp duomenų bazės dydžio ir įterpimo greičio iš dalies yra. Ši priklausomybė neaktuali MySQL duomenų bazės serveriui. Firebird ir PostgreSQL duomenų bazių serveriams matomas dėsningumas – kad didėjant duomenų bazei, ilgėja ir įterpimo laikas, bet tai nėra žymus pokytis.



18 pav. Įterpimas po 100 tūkst. eilučių

Įterpant po 100 tūkst., kaip ir 50 tūkst. eilučių į duomenų bazę greičiausiai atliko MySQL duomenų bazės serveris, jis vidutiniškai užtruko 86 sekundes. PostgreSQL užduotį atliko vidutiniškai per 237 sekundes, Firebird – per 306 sekundes. Taigi MySQL yra greitesnis beveik tris kartus už PostgreSQL ir daugiau nei 3,5 karto už Firebird.

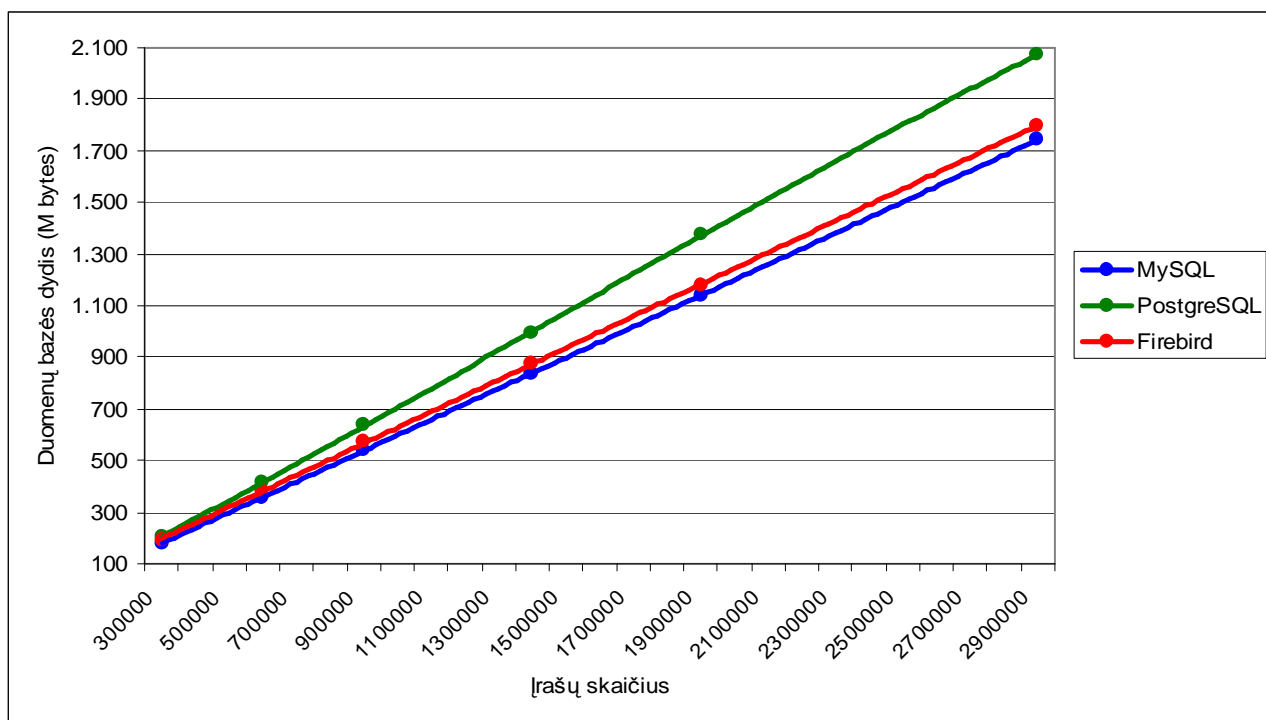
Įterpimo testas įterpant kaskart skirtingo dydžio duomenų kiekius pavaizduotas paveiksle nr. 19. Grafikas rodo, kad įterpimo greitis tiesiogiai priklauso nuo įterpiamų duomenų dydžio, t. y. jei 10 eilučių įterpimas užtruko vieną sekundę, 1000 eilučių įterpimas užtruks 100 kartų ilgiau, t. y. 100 sekundžių.



19 pav. Skirtingo dydžio duomenų įterpimas

Atlikus tyrimą įterpant į duomenų bazę skirtingo dydžio duomenis, pastebėta, kad MySQL duomenų bazės serveris atlieka daugiau nei du kartus greičiau nei Firebird ar PostgreSQL. Firebird yra lėčiausias įterpant duomenis, tačiau PostgreSQL yra tik nežymiai greitesnis.

Duomenų bazės dydžio ir įrašų joje tyrimas pavaizduotas paveiksle nr. 20. Įrašų kiekis tiesiogiai veikia duomenų bazės dydį, tolydžiai didėjant įrašų kiekiui, didėja ir duomenų bazės dydis.



20 pav. Įrašų skaičiaus ir duomenų bazės dydžio priklausomybė

Atminties vietą labiausiai taupo MySQL duomenų bazės serveris, jis iš nagrinėjamų atvirojo kodo duomenų bazių serverių nedaug aplenkė Firebird. PostgreSQL išnaudojo daugiausiai pastoviosios atminties.

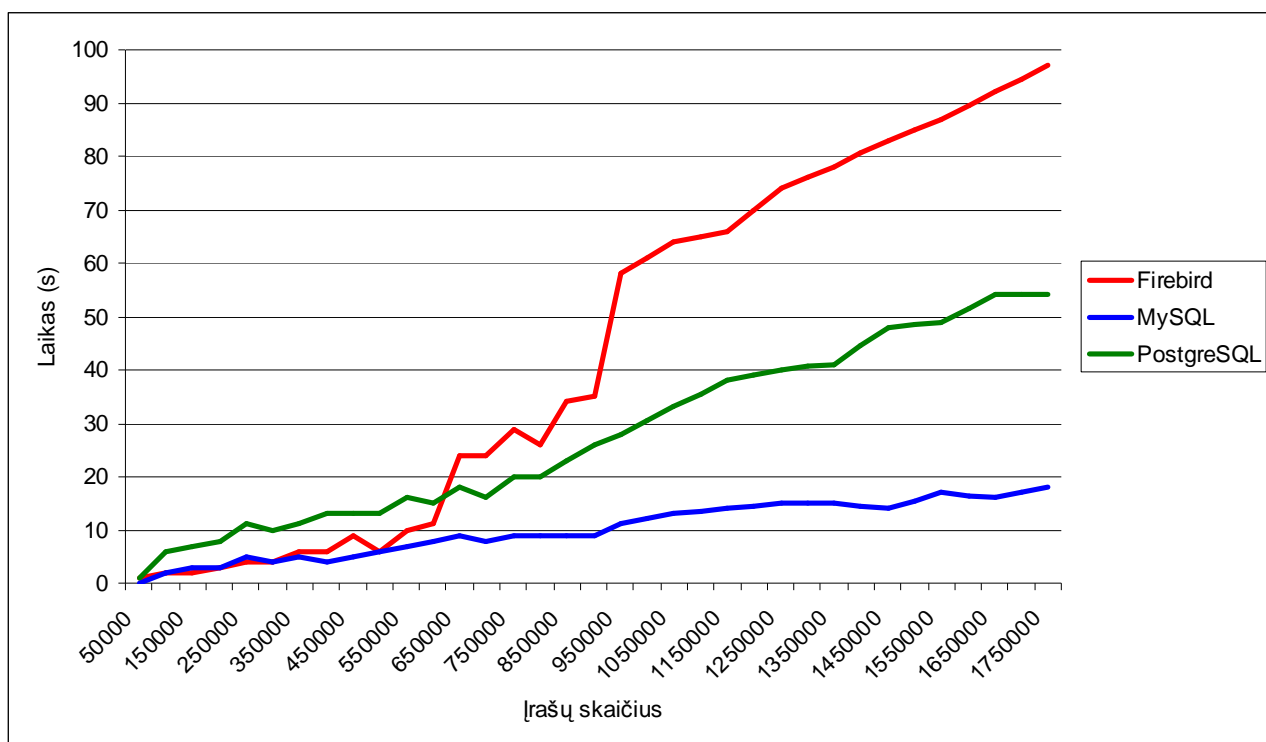
Išanalizavus įterpimo testų rezultatus galima teigti, kad geriausia yra pasirinkti MySQL duomenų bazės serverį. Jis buvo greičiausias visuose įterpimo testuose, taip pat taupiausiai išnaudojo pastoviąją atmintį. Priklausomai nuo to, kokie kriterijai yra svarbesni – pastovios vietos išnaudojimas ar įterpimo greitis – atitinkamai geriau pasirinkti Firebird ir PostgreSQL duomenų bazių serverius.

3.2. Koregavimo funkcijos tyrimas

Koregavimo funkcijai ištirti buvo panaudoti trys testai: koregavimas, nurodant įrašo ID, koregavimas, nurodant visą įrašo lauką ir koregavimas, nurodant įrašo lauko kontekstą. Testai buvo paleisti 35 kartus, su vis didesniu įrašų skaičiumi koreguojamoje lentelėje. Koregavimo, nurodant ID, testas atliko pakeitimus kas 100 lentelės įrašui. Taip buvo paruošta lentelė koregavimo, nurodant visą lauką, testui, kuris kaip tik koreguoja tuos įrašus, kuriuos pakeitė koregavimo, nurodant ID, testas. Taip pat koregavimas, nurodant lauko kontekstą, koreguoja tuos įrašus, kuriuos keitė koregavimo, nurodant visą lauką, testas.

Pagrindiniai koregavimai duomenų bazėje daromi, nurodant įrašo identifikacinį numerį, tokio tyrimo rezultatai pateikti paveiksle nr. 21. Grafike atsispindi, kad MySQL duomenų bazės serveris neturi lūžio taškų, kurią pasiekus, koregavimo, nurodant ID, laikas žymiai sumažėtų. PostgreSQL duomenų bazių serveris tokį lūžio tašką turi ties 900 000 įrašų koreguojamoje

lentelėje riba, tiesa, lūžis nėra toks didelis, kaip Firebird ties 650 000, kur koregavimo greitis padidėja beveik dvigubai.



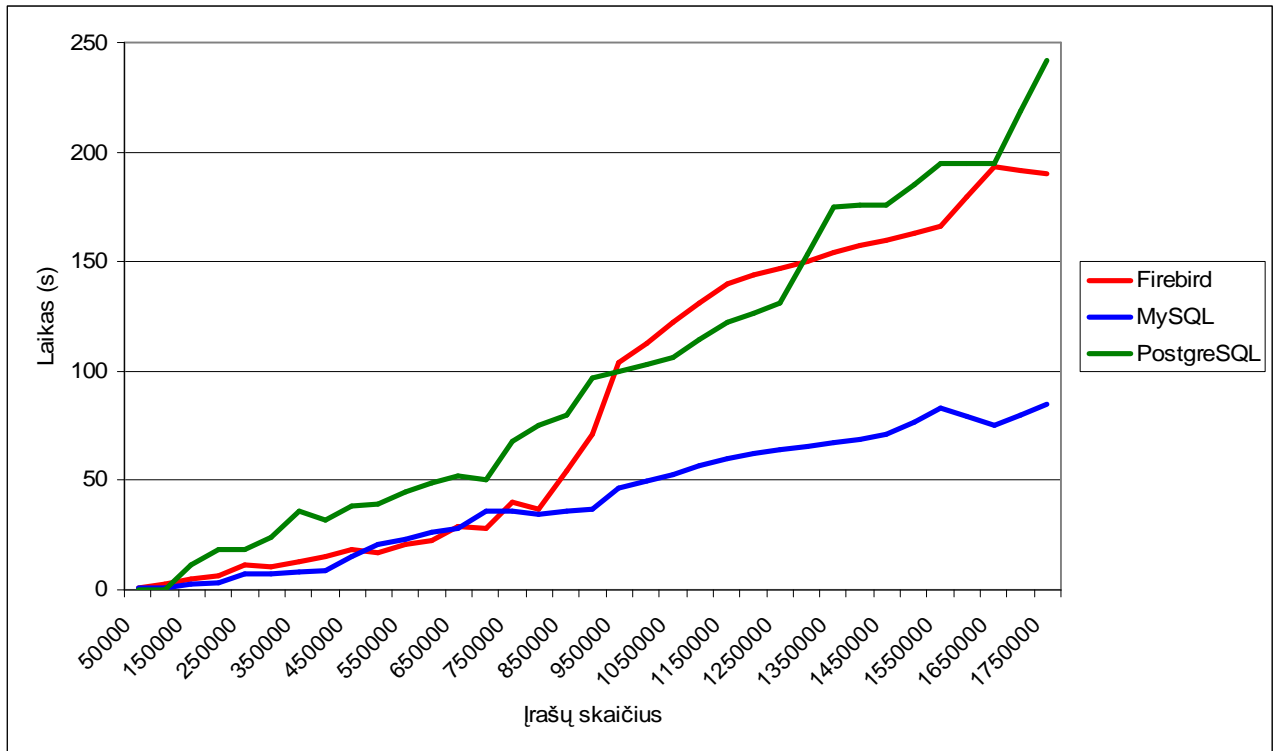
21 pav. Koregavimas, nurodant ID

MySQL duomenų bazės serveris koregavimą pagal ID atliko beveik trigubai greičiau už PostgreSQL, kuris turi nedidelį lūžio tašką, ir beveik penkis kartus greičiau nei stiprų lūžio tašką turintis Firebird. MySQL ir Firebird duomenų bazių serverių koregavimo pagal ID greitis yra labai panašus kol įrašų skaičius koreguojamoje lentelėje pasiekia 650 000 ribą.

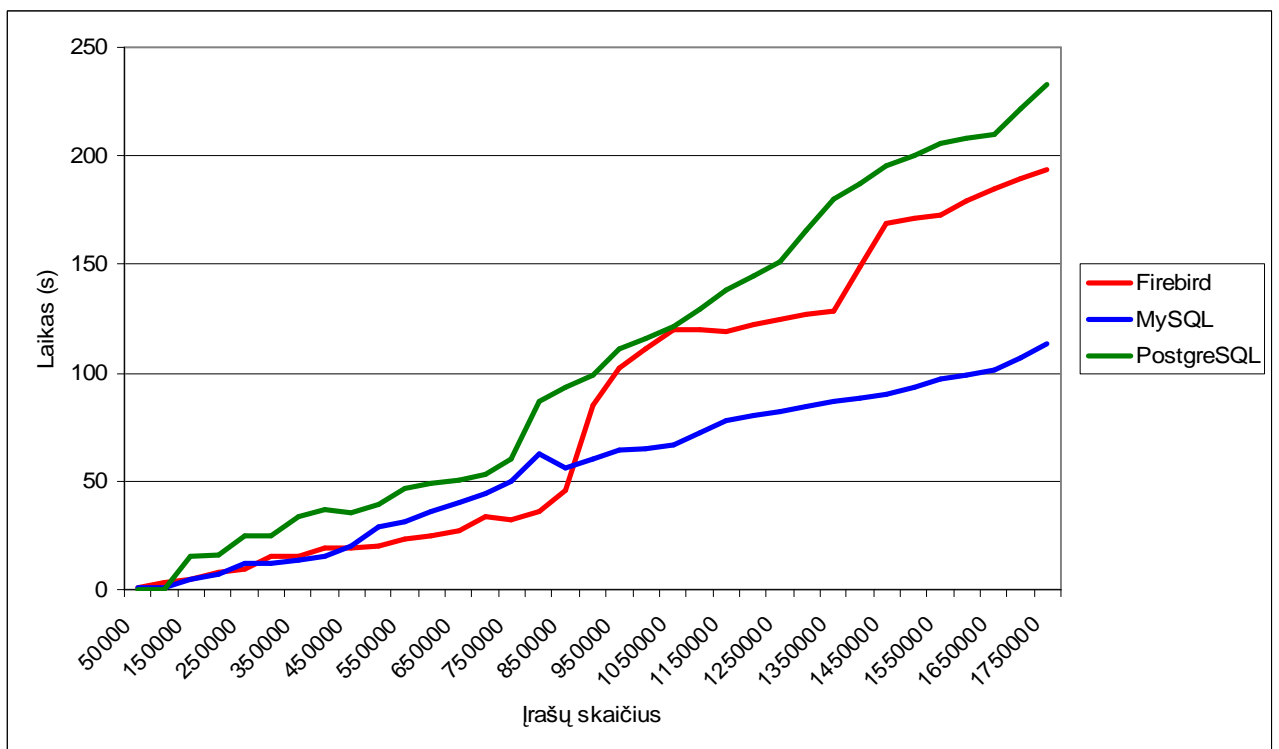
Tyrimo registruojant koregavimo, nurodant visą lauką, greitį rezultatai pateikti paveiksle nr. 22. esančiame grafike. Šiame tyrime, kaip ir koregavimo, nurodant ID tyrime, PostgreSQL ir Firebird duomenų bazių serveriai turi lūžio taškus, kuriuos pasiekus, koregavimo laikas labai padidėja.

MySQL duomenų bazės serveris sugaišta mažiau laiko nei Firebird ar PostgreSQL. MySQL taip pat neturi lūžio taškų. Kol koreguojama lentelė neviršija 850 000 įrašų ribos, MySQL ir Firebird duomenų bazės serveris koregavimą, nurodant visą lauką, atlieka per labai panašų laiko tarpą, tačiau peržengus 850 000 įrašų ribą, Firebird labai sulėtėja. PostgreSQL duomenų bazės serveris turi nedidelius lūžio taškus ties 800 000, 1 300 000 ir 1 700 000 įrašų koreguojamoje lentelėje ribos, kur sugaištas laikas padidėja šiek tiek daugiau.

Koregavimo, nurodant lauko kontekstą, tyrimo rezultatai pateikti paveiksle nr. 23. Šiame tyrime vėl pastebėti PostgreSQL ir Firebird duomenų bazių serverių lūžio taškai, kuriuos pasiekus, koregavimo, nurodant lauko kontekstą, laikas padidėja. Atliekant šį tyrimą, MySQL duomenų bazės serverio lūžio taškų, kur atlikimo laikas padidėja, nepastebėta.



22 pav. Koregavimas, nurodant visą lauką

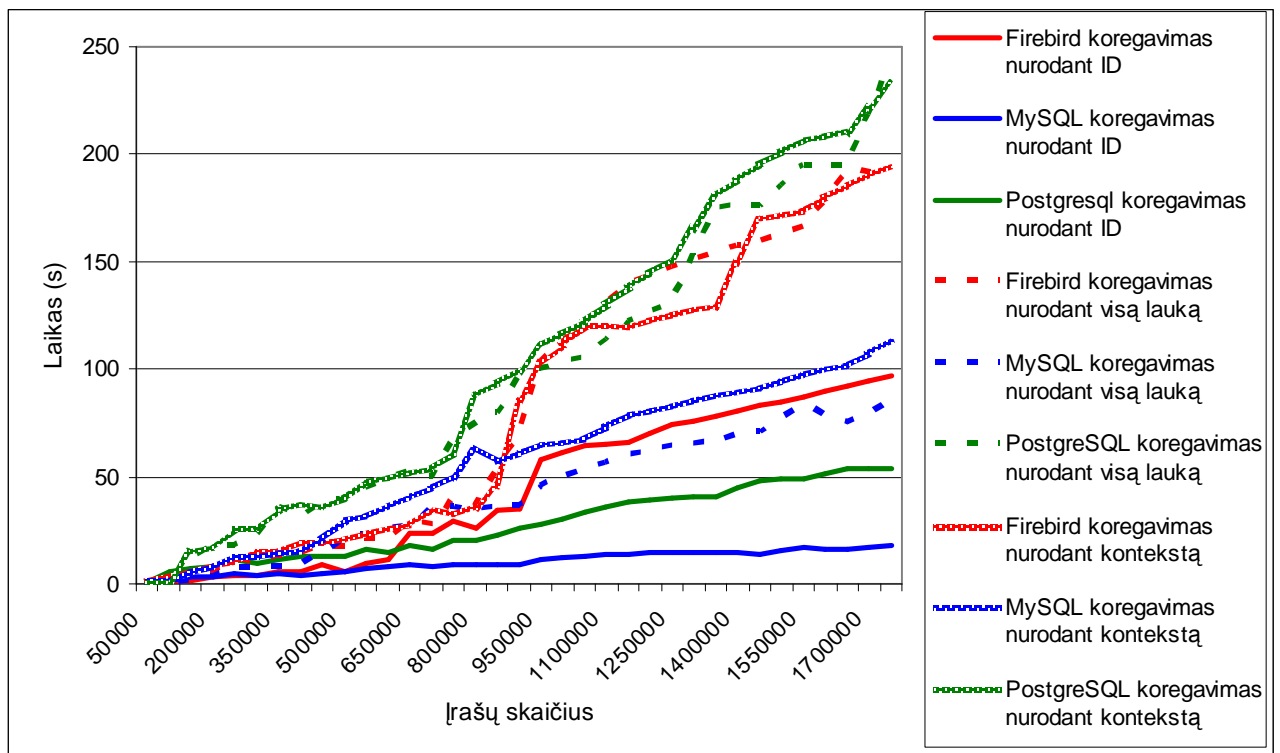


23 pav. Koregavimas, nurodant fragmentą

PostgreSQL duomenų bazės serveris, atliekant koregavimą, nurodant lauko kontekstą, sugaišo ilgiausiai iš nagrinėjamų objektų. Taip pat jis turėjo lūžio tašką ties 800 000 įrašų koreguojamoje lentelėje riba, po kurios labai atsiliko nuo MySQL duomenų bazės serverio.

Analizuojant koregavimą, nurodant lauko kontekstą, kai koreguojama lentelė turi iki 850 000 įrašų, greičiausias yra Firebird duomenų bazės serveris, tačiau po lūžio ties minėta riba stipriai sulėtėja ir MySQL duomenų bazės serveris, kuris neturi lūžio taško, yra greitesnis.

Visi trys tyrimai susiję su koregavimo funkcija koregavo tuos pačius įrašus, t. y. kiekvienas testas duomenų bazėje atliko identiškus veiksmus. Paveikslas nr. 24 vaizduoja visų trijų testų duomenis viename grafike, todėl nesunkiai galima pasakyti, kuris būdas koreguoti įrašus yra greičiausias.



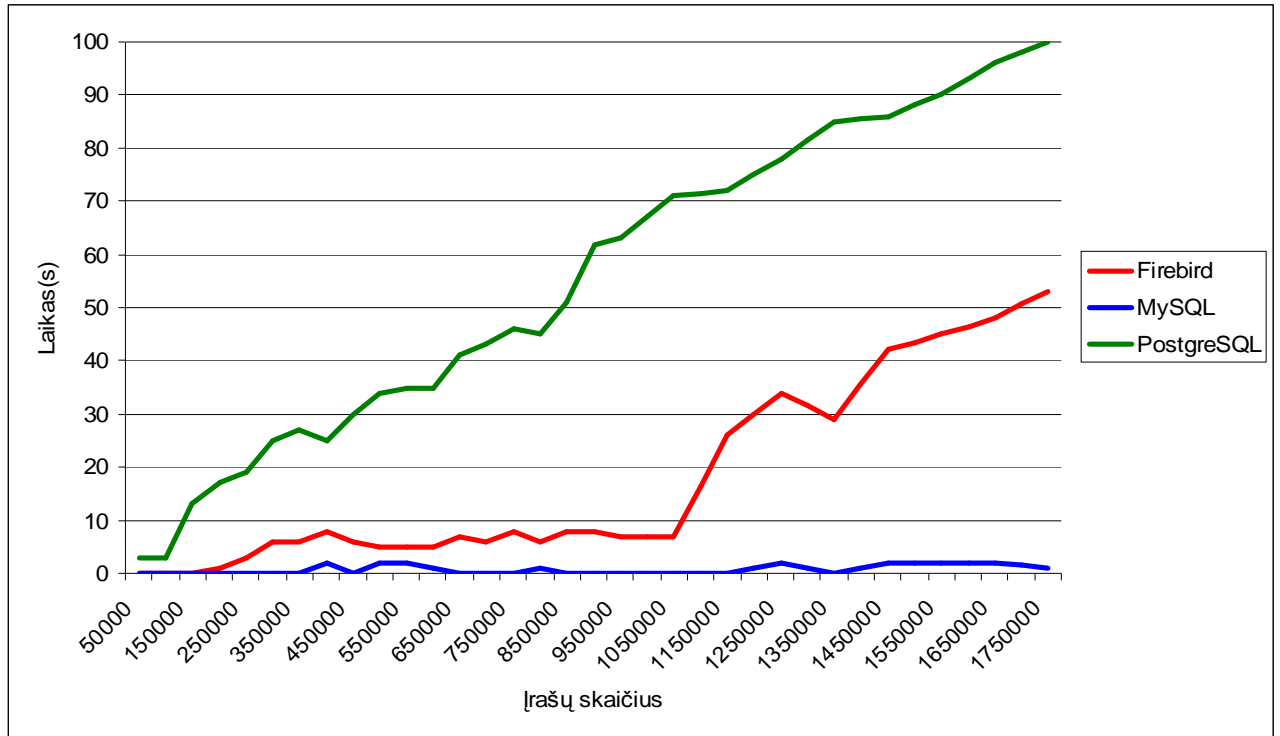
24 pav. Koregavimas, nurodant ID, visą lauką ir fragmentą

MySQL duomenų bazės serveris yra greičiausias, atliekant visus tris su koregavimu susijusius tyrimus, jis neturi lūžio taškų. PostgreSQL duomenų bazės serveris visuose trijuose tyrimuose turėjo lūžio taškų, kuriuose laikas padidėjo, jis lieka lėčiausias, išskyrus atliekant koregavimą, nurodant įrašo ID, ir kai koreguojama lentelė turi daugiau nei 650 000 įrašų. Firebird duomenų bazės serveris yra labai greitas tik tada, kai koreguojama lentelė turi iki 650 000 įrašų, peržengus šią ribą, greitis labai sumažėja.

3.3. Paieškos funkcijos tyrimas

Paieškos tyrimui buvo sukurti du testai: viename paieška vykdoma, nurodant visą ieškomo lauko tekstą, kita – nurodant ieškomo lauko fragmentą. Abu paieškos testai ieško tokių pačių įrašų, kuriuos pakeitė koregavimo testai. Abu testai buvo kartojami 35 kartus, kiekvieną kartą vis didesnėje duomenų bazėje.

Paieškos tyrimas, nurodant visą ieškomo lauko tekstą, pavaizduotas paveiksle nr. 25. Paieška tiesiogiai priklauso nuo duomenų bazės dydžio – kuo didesnė duomenų bazė, tuo ilgiau užtrunka paieška. To negalima pritaikyti MySQL duomenų bazės serveriui, jo paieškos greitis pagal visą lauką nepakito viso tyrimo metu.

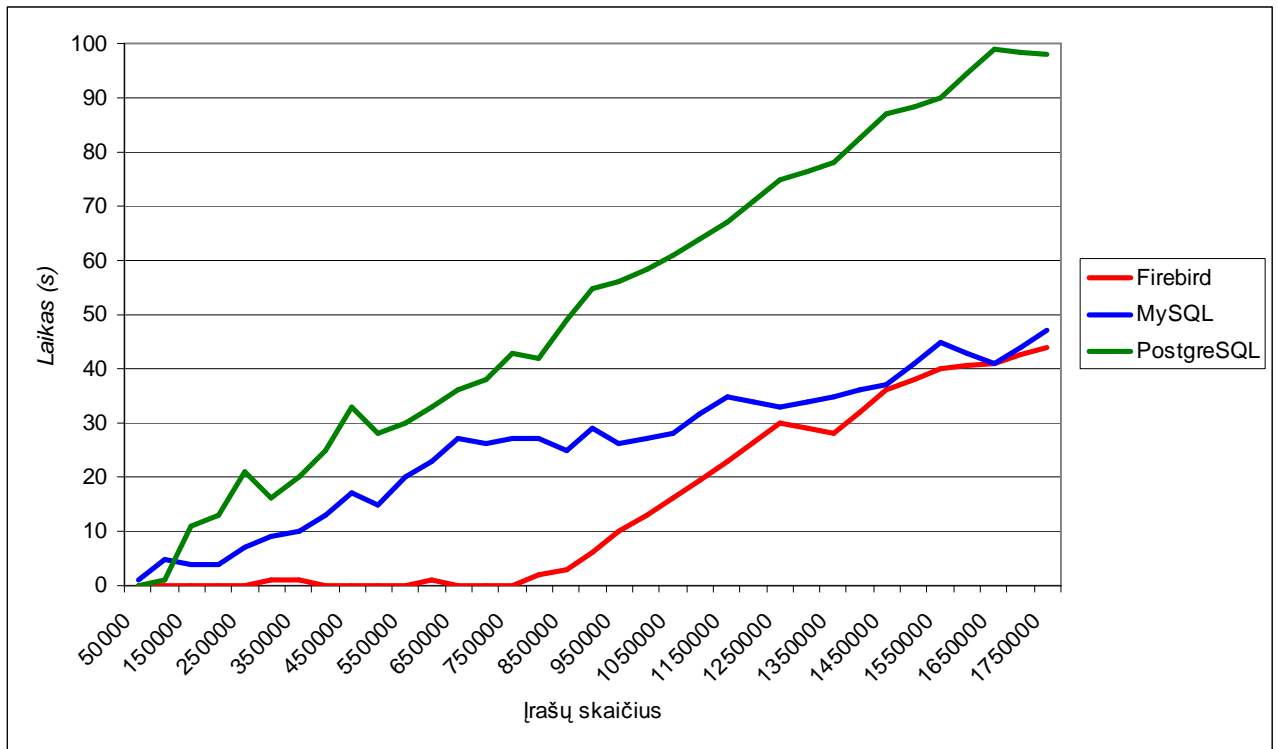


25 pav. Paieška, nurodant visą lauką

Iš paveikslo nr. 25 matoma, kad MySQL paieškos, nurodant visą lauką, užtrunka iki 2 sekundžių ir nepriklauso nuo įrašų skaičiaus, kuriuose vykdoma paieška. Firebird matomas greičio kritimas, kai įrašų skaičius ieškomoje lentelėje pasiekia 1 100 000 ribą. Firebird lieka antras, suradęs reikiamus įrašus vidutiniškai dvigubai greičiau nei PostgreSQL duomenų bazės serveris.

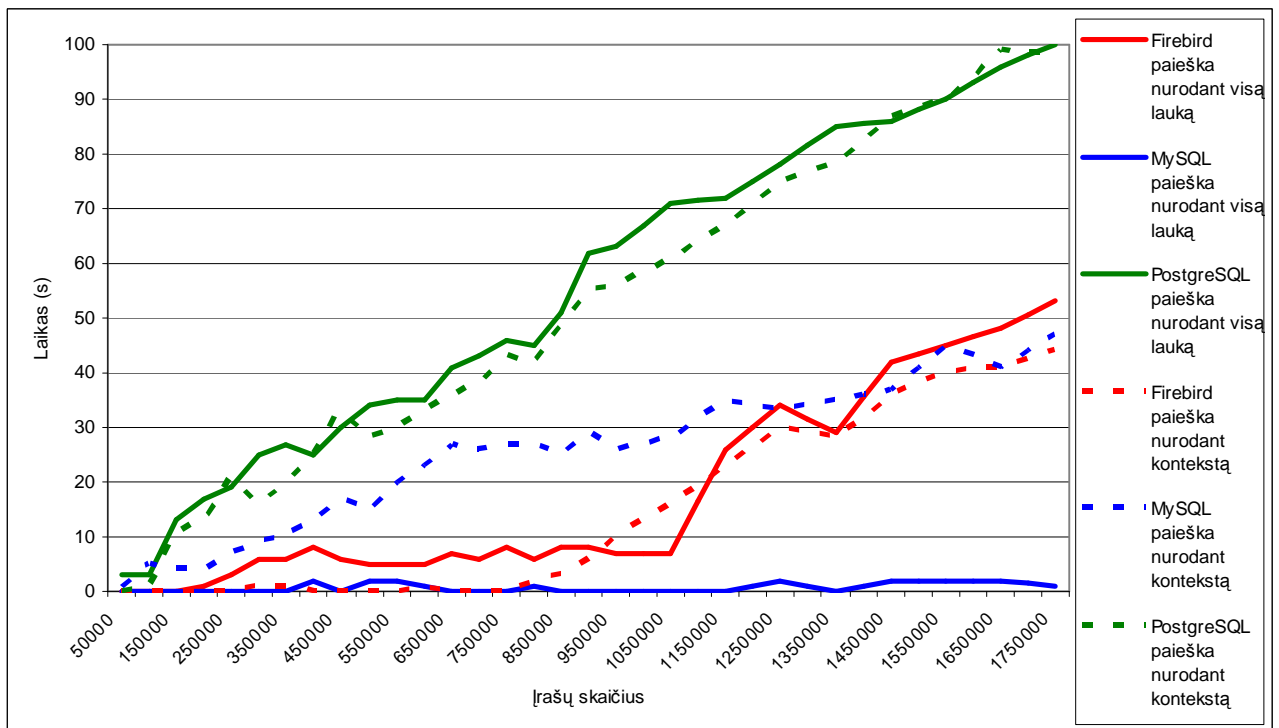
Paieškos pagal kontekstą rezultatai pateikti paveiksle nr. 26. Atliekant paiešką pagal kontekstą, buvo užduodamas tas pats laukas, kaip ir paieškai, nurodant visą lauką, taigi surastų duomenų kiekiai sutapo tiek naudojant paiešką pagal visą lauką, tiek pagal kontekstą.

Greičiausiai paiešką pagal kontekstą atliko Firebird duomenų bazės serveris. Paieškoje pagal visą lauką jis turi lūžio tašką, nuo kurio paieškos greitis krenta. Firebird duomenų bazės serverio paieškos pagal kontekstą greičio lūžio taškas yra ties 800 000 įrašų riba. MySQL turi 650 000 įrašų lentelėje ribą, nuo kurios paieškos pagal kontekstą laikas nebedidėja taip sparčiai, kaip iki jos. PostgreSQL paieškos pagal kontekstą užtruko ilgiausiai.



26 pav. Paieška, nurodant fragmentą

Sudėjus abi paieškas į vieną grafiką, kas matoma paveiksle nr. 27, galima lengvai pastebėti duomenų bazių serverių paieškos pagal visą lauką ir kontekstą skirtumus.



27 pav. Paieška, nurodant visą lauką ir fragmentą

PostgreSQL ir Firebird duomenų bazių serveriai paiešką pagal kontekstą atlieka šiek tiek greičiau nei paiešką pagal visą lauką. MySQL duomenų bazės serveris, priešingai nei

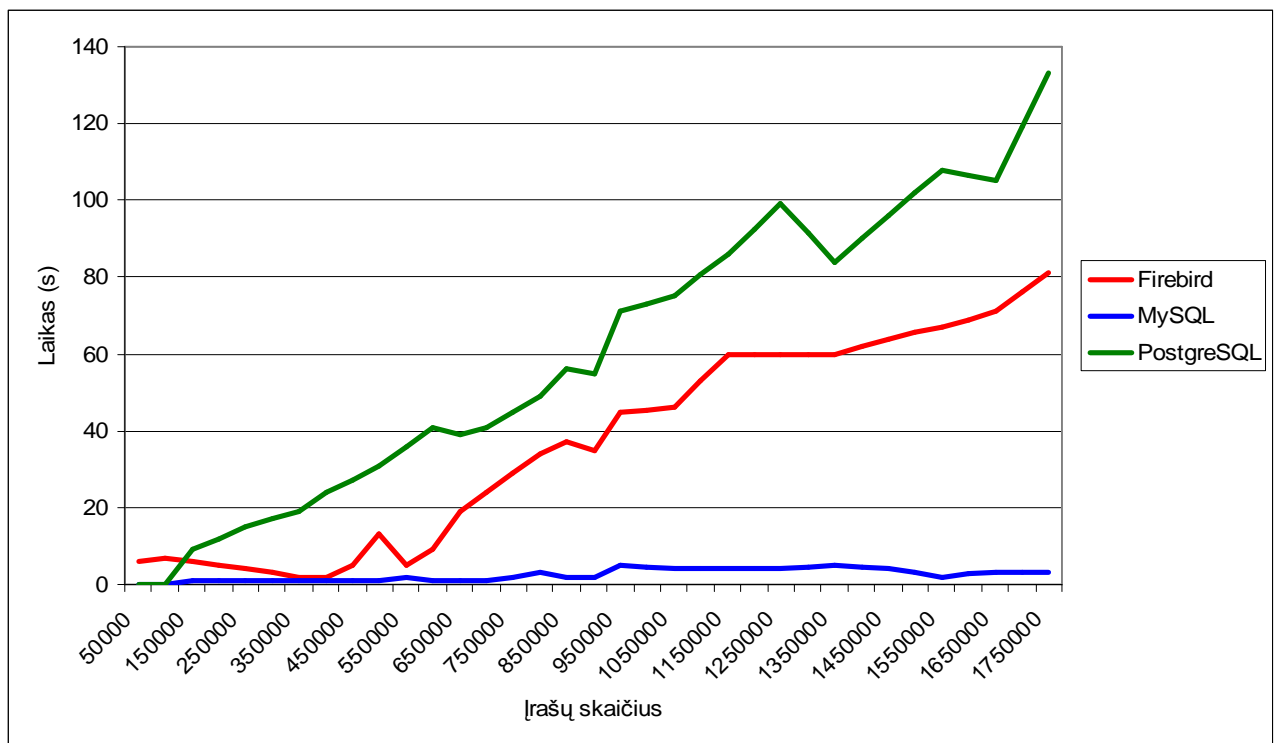
Firebird ir PostgreSQL, greičiau atlieka paiešką pagal visą lauką nei pagal kontekstą, ir šis skirtumas yra ryškus.

Vertinant duomenų bazių serverius pagal paieškos įvykdymo kriterijų lėčiausias iš nagrinėjamų duomenų bazių serverių yra PostgreSQL. Jei pagrindinis kriterijus būtų paieška pagal visą lauką – greičiausias MySQL, jei kriterijus būtų paieška pagal kontekstą – greičiausias Firebird. Jei svarbios abi paieškos ir įrašų skaičius ieškomoje lentelėje neviršija 1 100 000 ribos – greitesnis yra Firebird duomenų bazės serveris, jei įrašų skaičius daug didesnis, greitesnis bus MySQL.

3.4. Trynimo funkcijos tyrimas

Trynimo funkcijai iširti buvo naudojami du testai: trynimas, nurodant visą lauką, ir trynimas, nurodant lauko kontekstą. Trynimo testai trynė koregavimo, nurodant ID, testų padarytus pakeitimus, todėl tam, kad antrasis testas turėtų ką trinti, teko paleisti koregavimo, nurodant ID, testą pakartotinai. Testai buvo paleisti 35 kartus, kaskart trinant didesnę įrašų skaičių vis daugiau įrašų turinčioje lentelėje.

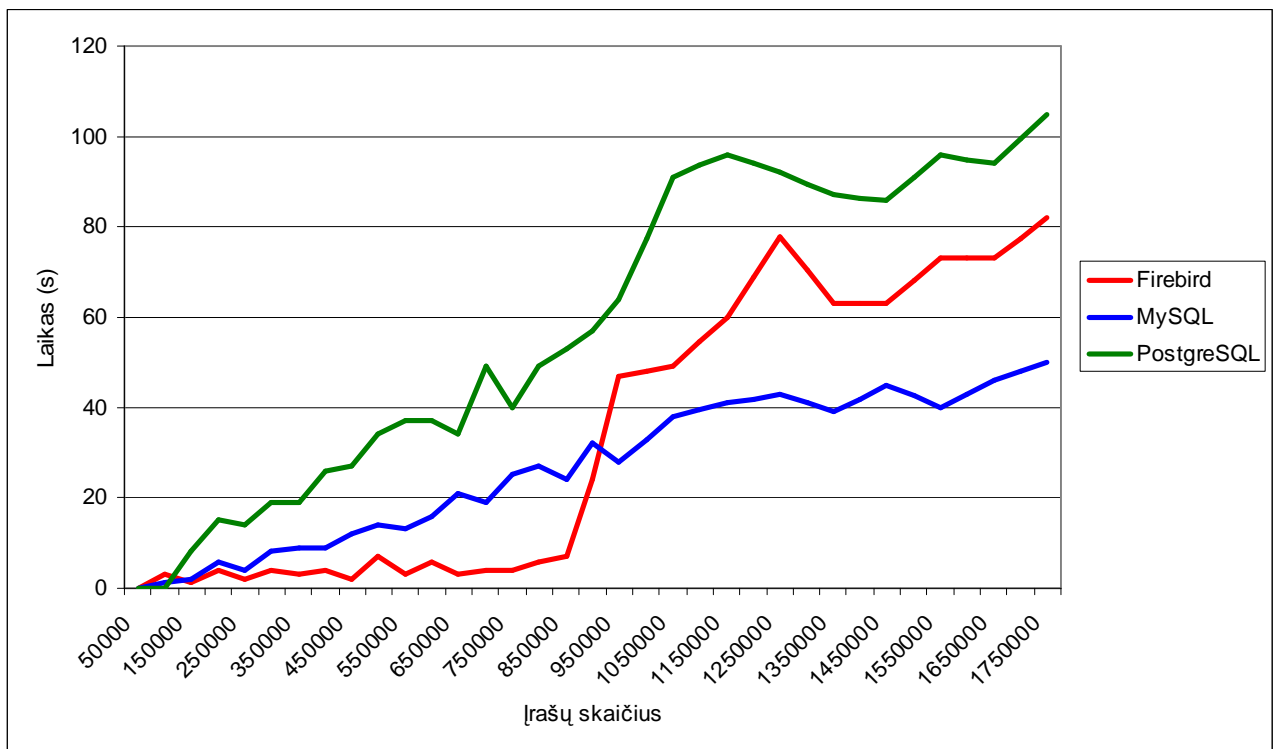
Trynimo, nurodant visą lauką, tyrimo rezultatai pateikti paveiksle nr.28. Su PostgreSQL ir MySQL duomenų bazių serveriais atliktas tyrimas neparodė lūžio taškų, kuriuos pasiekus, trynimo greitis labiau sumažėtų. Firebird duomenų bazės serveris turi lūžio tašką, serverio atliekamas trynimas, nurodant visą lauką, labiau sulėtėjo lentelėje, turinčioje daugiau nei 650 000 įrašų.



28 pav. Trynimas, nurodant visą lauką

Trynimo, nurodant visą lauką, tyrimas parodė, kad MySQL duomenų bazės serveris buvo greičiausias iš nagrinėjamų duomenų bazės serverių, taip pat trinamos lentelės dydis mažai veikia sugaištą trynimui laiką. Firebird duomenų bazės serveris pajėgus konkuruoti su MySQL duomenų bazės serveriu, jei trinama lentelėje neviršijant 650 000 įrašų ribos, už šios ribos Firebird duomenų bazės serveris labai sulėtėja. PostgreSQL duomenų bazės serveris, nors ir neturi akivaizdžių sulėtėjimo taškų, viso trynimo, nurodant visą lauką, tyrimo metu buvo lėčiausias.

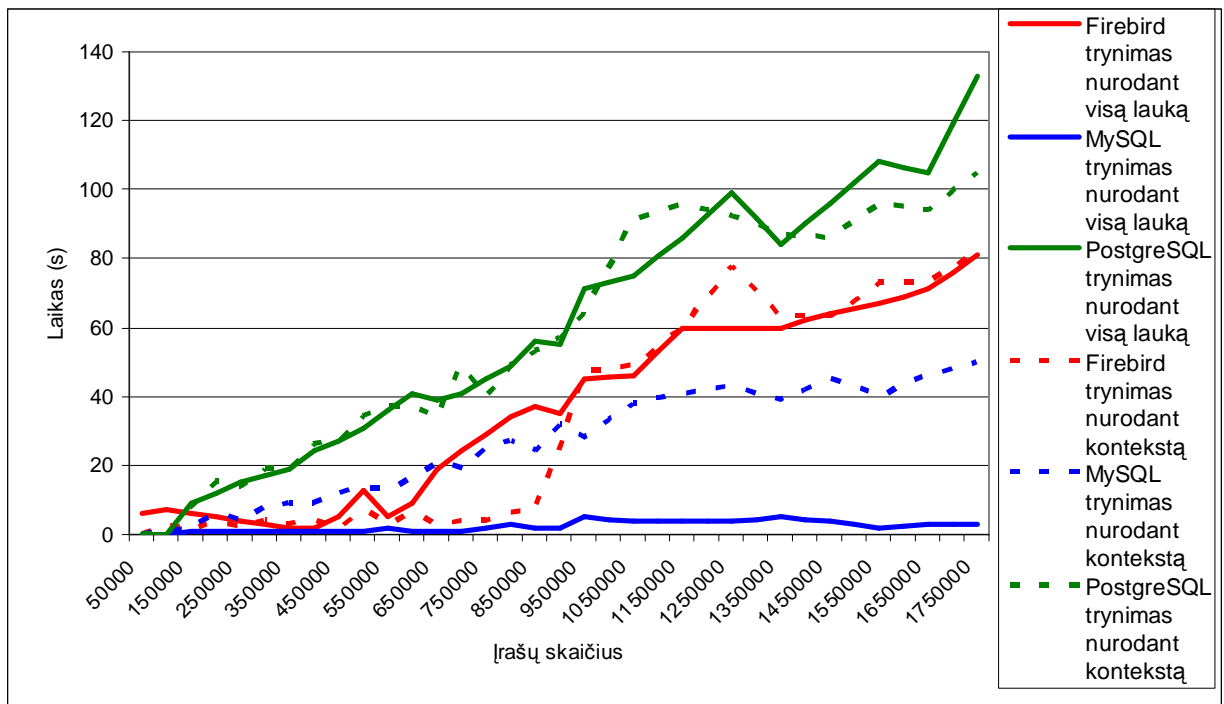
Trynimo, nurodant lauko kontekstą, rezultatai pateikti paveiksle nr. 29. Tyrimas parodė Firebird duomenų bazės serverio sulėtėjimo tašką ties 950 000 įrašų trinamoje lentelėje. PostgreSQL duomenų bazės serveris taip pat turi sulėtėjimo tašką ties 950 000 įrašų trinamoje lentelėje, tačiau pasiekus 1 500 000 įrašų ribą, serveris darbą atlieka tuo pačiu dėsningumu, kaip iki sulėtėjimo taško.



29 pav. Trynimas, nurodant fragmentą

Firebird duomenų bazės serveris, atliekant trynimą, nurodant lauko kontekstą, yra greičiausias lentelėje, turinčioje iki 900 000 įrašų, didesnėje lentelėje serveris labai sulėtėja ir tampa lėtesnis nei MySQL duomenų bazės serveris. PostgreSQL duomenų bazės serveris šio tyrimo metu buvo lėčiausias.

Abu su trynimu susiję testai trynė tą patį duomenų kiekį tokio pat dydžio lentelėse, t. y. kiekvienas testas duomenų bazėje atliko identiškus veiksmus. Paveikslas nr. 30. vaizduoja abu tyrimus viename grafike, taip nesunkai matoma, kuriam duomenų serveriui patogiau atlikti trynimą, nurodant lauką ar lauko kontekstą.

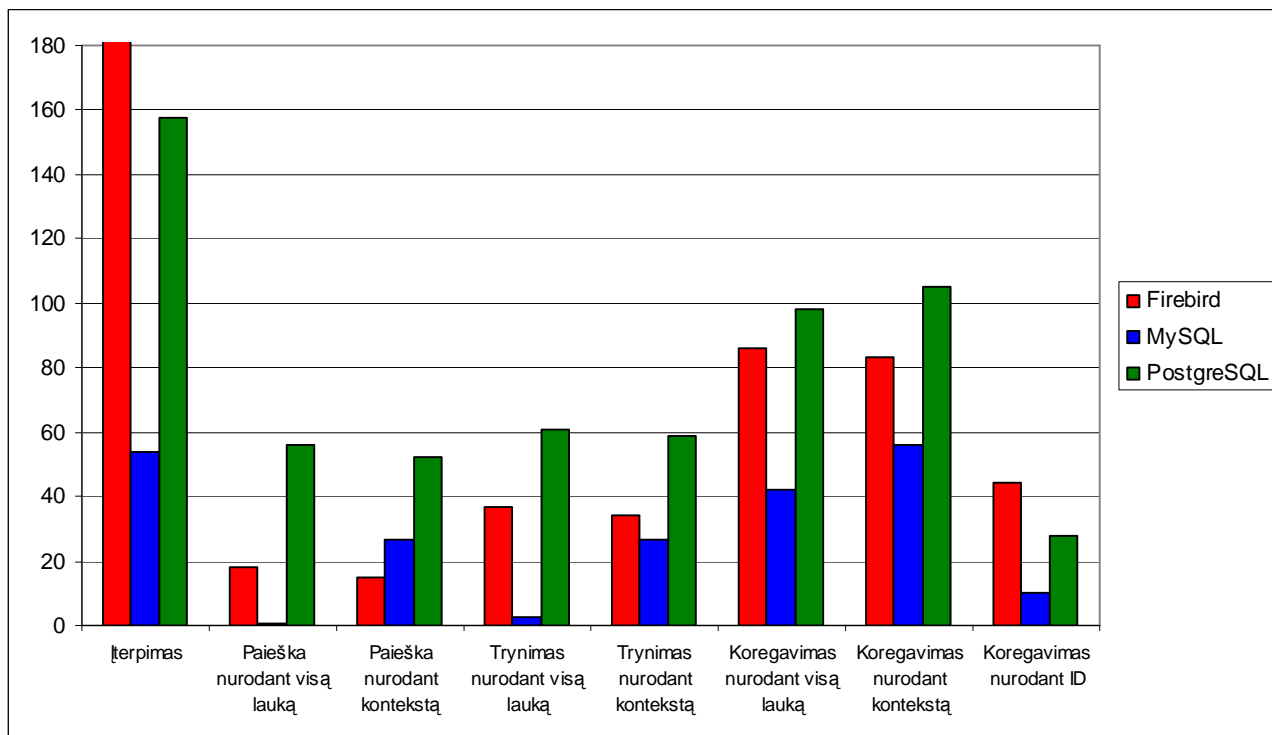


30 pav. Trynimas, nurodant visą lauką ir fragmentą

PostgreSQL ir Firebird duomenų bazių serveriai, atliekant trynimą tiek nurodant visą lauką, tiek lauko kontekstą, sugaišta labai panašių laikų, tačiau Firebird duomenų bazės serveris, atliekant trynimą, nurodant visą lauką turi sulėtėjimo tašką ties 650 000 įrašų trinamoje lentelėje riba, o atliekant trynimą, nurodant lauko kontekstą, sulėtėjimo taškas yra ties 900 000 įrašų riba. MySQL duomenų bazės serverio trynimo tyrimai labai skiriasi, šis serveris trynimą visada atliks greičiau, nurodant visą lauką.

3.5 Tyrimo rezultatų suvestinė

Tyrimo metu buvo tiriamos keturios duomenų bazių savybės: įterpimas, paieška, trynimas ir koregavimas. Tokios savybės kaip paieška, trynimas ar koregavimas buvo tiriamos kelias atskirais tyrimais. Viso tyrimo metu gautų duomenų vidurkiai pateikti grafiškai paveiksle nr. 31. Aptarnavimo paslaugų teikimo valdymo sistema vargu ar turės duomenų bazę su lentelėmis, kurių įrašų skaičius viršys 500 000 ribą, todėl paveikslas nr. 31 pateikia vidurkius dirbant su lentelėmis iki 500 000 įrašų.

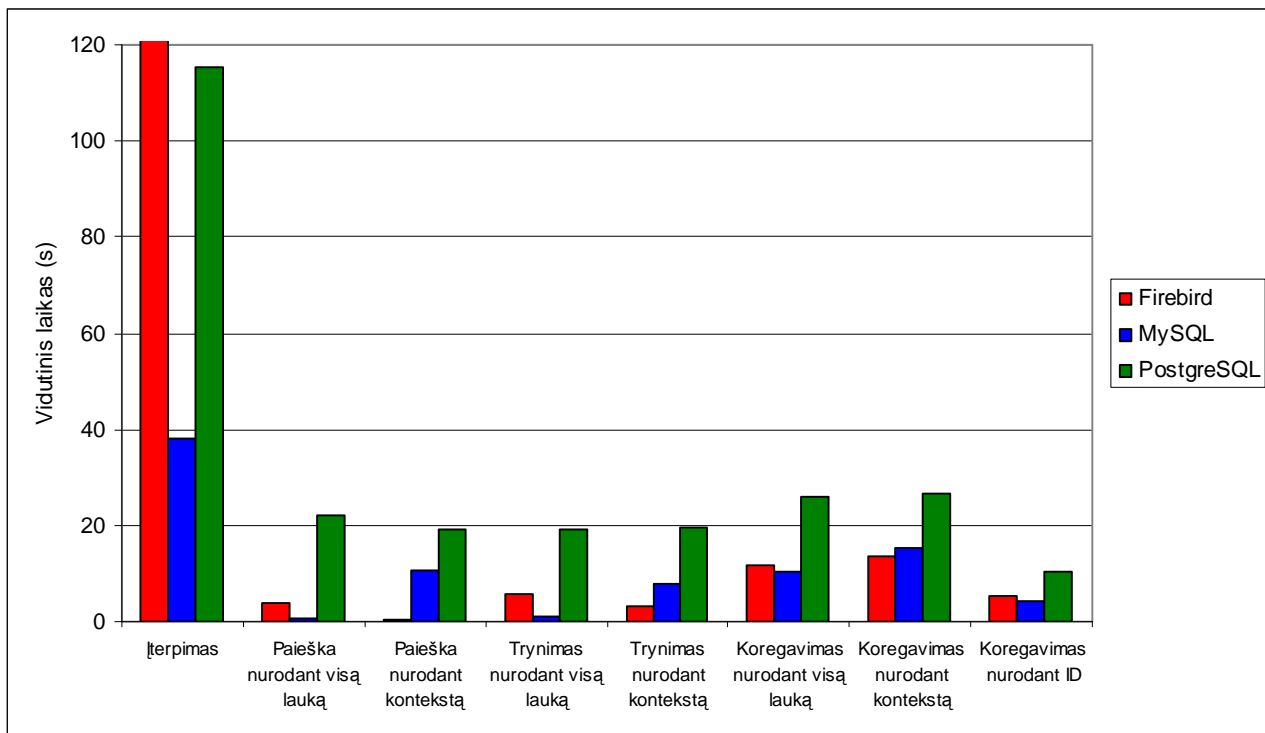


31 pav. Rezultatų suvestinė

Viso tyrimo metu gautų vidurkių grafikas rodo, kad 7 iš 8 testų MySQL duomenų bazės serveris buvo greičiausias, taip pat MySQL duomenų bazės serveris išnaudojo mažiausiai pastovios atminties duomenų bazei saugoti. Firebird duomenų bazės serveris paiešką, nurodant kontekstą, atliko greičiausiai ir 6 tyrimuose buvo greitesnis už PostgreSQL duomenų bazės serverį, kuris lieka lėčiausias iš nagrinėjamų duomenų bazių serverių.

Pasirinkus duomenų bazę iki 500 000 įrašų vienoje lentelėje, rezultatai skiriasi nuo rezultatų, atlikus tyrimus su duomenų baze, turinčia 1 800 000 įrašų vienoje lentelėje. Aptarnavimo paslaugų teikimo valdymo sistema tokio įrašų skaičiaus vienoje lentelėje greičiausiai neturės, todėl paveiksle nr. 32. pateikti rezultatai yra artimesni realiam sistemos darbui, t. y. įrašų skaičius vienoje lentelėje neperžengia 500 000 ribos.

Firebird duomenų bazės serveris, atliekant tyrimus su duomenų baze, turinčia iki 500 000 įrašų vienoje lentelėje, yra greičiausias trijuose tyrimuose, MySQL duomenų bazės serveris greitesnis penkiuose tyrimuose, PostgreSQL duomenų bazės serveris yra lėčiausias, išskyrus įterpimo testavimą.



32 pav. Rezultatų suvestinė, kai lentelės dydis neviršija 500 000 įrašų

Santrauka (anglų kalba)

Jankevičius, Vytautas (2006). “Analysis of open source database servers using service management system”. MA Graduation Paper. Kaunas: Faculty of informatics, University of Kaunas Technology . 49 p.

The graduation work object is to analyse the popular open source databases servers and determine which is the optimal for service management system.

To complete the goal there were made such tasks:

- Perform theoretical analysis of open source database servers
- To determine similarities and differences of open source database servers.
- To analyze and compare functional features of service management systems.
- To design the special class and methods necessary for research.
- To determine the optimal database server for service management system using the special class and methods.

There are 49 pages in the work. First part of the work consists of 13 pages, second part – 13 pages and the third part is 14 pages. There are 3 tables and 32 pictures in the work.

The first part of the work proposed the review and the comparison of service management systems and open source database servers.

Second part of the work is proposed the project model of service management system

Third part of the work proposed the results of open source database servers’ research.

.

Terminų ir santrumpų žodynas

Administratorius - visateisis kompiuterio, kompiuterių tinklo arba kurios nors kitos kompiuterių sistemos tvarkytojas. Rūpinasi kasdiene priežiūra, atnaujinimu, saugumu ir kitais su jo administruojama sistema susijusiais klausimais

CPU - Centrinis procesorius (Central Processing Unit)

DB - Duomenų bazė.

DBVS - Duomenų Bazių Valdymo Sistema

GUI - Grafinė vartotojo sąsaja (Graphical User Interface)

HTML - Žiniatinkliu perduodamo hiper-teksto programavimo kalba (Hyper Text Markup Language)

HTTP - Duomenų perdavimo tinklu protokolas (hyper transfer transmission protocol).

Informacinė sistema - tai informacijos apdorojimo sistema kartu su atitinkamais žmonių, techninių priemonių, finansų resursais, kurianti ir skirstanti informacija.

IT - Informacinės technologijos (Information Technologies)

OS - Operacinė sistema, programų komplektas, valdantis aparatinės įrangos darbą ir teikianti sąsają su kompiuterio naudotoju. OS yra terpė, kurioje gali veikti kitos, į ją įdiegtos, programos.

PC - Personalinis kompiuteris (Personal Computer)

PĮ - Programinė įranga

Programuotojas – asmuo, vykdamas programinių modulių įgyvendinimą pagal pateiktas specifikacijas.

Projektų vadovas – žmogus, atsakingas už projekto kūrimo eigos stebėjimą, projekto proceso valdymą, darbų bei resursų paskirstymą.

RAM - Operatyvinė atmintis (Random Access Memory)

Sistemos gyvavimo ciklas - sistemos plėtros pokyčių nuo koncepcijos iki veikimo pabaigos visuma.

Testavimo objektas - Komponentė(-s), taikomoji programa, sistema ir kt., t.y. tai, kas turi būti testuojama

Testuotojas – atsitiktine tvarka parinkti įvairių profilių žmonės.

UIS - Vartotojo sąsajos specifikacija (User Interface Specification)

UML - Programinės įrangos projektavimo kalba (Unified Modeling Language)

USE CASE - Panaudojimo atvejis

Užsakovas – užsakovo įmonės atstovas.

Vartotojas – žmogus naudosis šią sistemą.

WWW - Pasaulinis Žiniatinklis (World Wide Web).

Išvados

1. Atlikus atvirojo kodo duomenų bazių serverių analizę galima teigti, kad visi trys aptarti duomenų bazių serveriai yra labai panašūs skiriasi tik kelios funkcijos arba jos vadinamos skirtingais vardais, nors atlieka tuos pačius veiksmus. MySQL ir PostgreSQL duomenų bazių serveriai turi spartinančios atminties galimybes, Firebird tokios galimybės neturi.

- *Transakcijos.* Tokios ypatybės kaip ACID standartas, dviejų fazių patvirtinimo protokolas, konkurencijos valdymas ar duomenų kopijos ir atstatymas yra palaikomos visų analizuotų duomenų bazių serverių.

- *Valdymas.* Visi analizuoti duomenų bazių serveriai turi galimybę saugoti ir paleisti vartotojo kurtas funkcijas, taip pat kaip standartas palaikomi trigeriai. MySQL ir PostgreSQL turi galimybę valdyti spartinančią atmintį.

- *Architektūra.* Visi analizuoti duomenų bazių serveriai gali būti įdiegti įvairaus tipo sistemose. MySQL ir Firebird turi po keletą serverio variantų, o PostgreSQL turi tik vieną variantą.

- Vienintelis MySQL palaiko SQL99 standartą, kai PostgreSQL ir Firebird palaiko SQL 2003 standartą

2. Aptartos keturios sistemos savo funkcinėmis galimybėmis labai panašios, tačiau skirtumų yra. Registruojant prekes ELKO, Servicenet ir GNT naudoją vieną langą, tuo tarpu ACME – tris langus. Prekės pristatymą galima pasirinkti trijose sistemose: GNT, ACME ir Servicenet. ELKO vienintelis reikalauja nurodyti sąskaitos faktūros numerį, kurios pagrindu buvo pirktą prekę, kitose sistemose tai nurodomai tik tuo atveju kai prekę nėra randama duomenų bazėje. ACME vieninteliai naudojami penkiomis fiksuotomis prekių būsenomis, kitų sistemų būsenos nurodomos įmonės darbuotojų. ACME sąrašas suskirstytas į grupes pagal esamas būsenas, panaudotas medžio tipas

3. Suprojektuota specialiai tyrimui reikalinga klasė ir metodai buvo panaudoti atliekant tyrimus, kurių metu nustatyta:

- Išanalizavus įterpimo testų rezultatus galima teigti, kad geriausia pasirinkti MySQL duomenų bazės serveri, jis buvo greičiausias visuose įterpimo testuose, taip pat taupiausiai išnaudojo pastoviąją atmintį. Priklausomai nuo to kokie kriterijai yra svarbesni, ar pastovios vietos išnaudojimas ar įterpimo greitis, atitinkamai geriau pasirinkti Firebird arba PostgreSQL duomenų bazių serverį.

- MySQL duomenų bazės serveris yra greičiausias atliekant visus tris su koregavimu susijusius tyrimus, jis neturi lūžio taškų. PostgreSQL duomenų bazės serveris visuose trijuose tyrimuose turėjo lūžio taškų, kuriuose laikas padidėjo, jis lieka lėčiausias išskyrus atliekant koregavimą nurodant įrašo ID ir kai koreguojama lentelė turi daugiau nei 650 000 įrašų. Firebird duomenų bazės serveris yra labai greitas tik tada kai koreguojama lentelė turi iki 650 000 įrašų, peržengus šią ribą greitis ženkliai sumažėja.

- Vertinant duomenų bazių serverius pagal paieškos įvykdymo kriterijų lėčiausias iš nagrinėjamų duomenų bazių serverių yra PostgreSQL. Jei pagrindinis kriterijus paieška pagal visą lauką - greičiausias MySQL, jei kriterijus paieška pagal kontekstą greičiausias Firebird.

- PostgreSQL ir Firebird duomenų bazių serveriai atliekant trynimą, tiek nurodant visą lauką, tiek lauko kontekstą, sugaištą labai panašius laikus, tačiau Firebird duomenų bazės serveris turi lūžio taškus. MySQL duomenų bazės serverio trynimo tyrimai skiriasi ženkliai, šis serveris trynima visada atliks greičiau nurodant visą lauką.

- Viso tyrimo metu gautų vidurkių grafikas rodo, kad 7 iš 8 testų MySQL duomenų bazės serveris buvo greičiausias, taip pat MySQL duomenų bazės serveris mažiausiai išnaudojo pastovios atminties duomenų bazei saugoti. Firebird duomenų bazės serveris paiešką nurodant kontekstą atliko greičiausiai ir 6 tyrimuose buvo greitesnis už PostgreSQL duomenų bazės serverį, kuris bendrame kontekste lieka lėčiausias iš nagrinėjamų duomenų bazių serverių.

- Firebird duomenų bazės serveris atliekant tyrimus su duomenų baze, turinčia iki 500 000 įrašų vienoje lentelėje yra greičiausias trijuose tyrimuose, MySQL duomenų bazės serveris greitesnis penkiuose tyrimuose, PostgreSQL duomenų bazės serveris išskyrus įterpimo testavimą yra lėčiausias.

4. Pristatytas straipsnis „Atvirojo kodo duomenų bazių serverių analizė“ 11-oje tarpuniversitetinėje doktorantų magistrantų konferencijoje, informacinės technologijos 2006.[13]

Literatūra

1. MySQL AB: The world's most popular open source database. [interaktyvus]. [žiūrėta 2006-02-12]. Prieiga per internetą: www.mysql.com.
2. PostgreSQL: The world's most advanced open source database [interaktyvus]. [žiūrėta 2006-02-12]. Prieiga per internetą: www.postgresql.org.
3. Firebird - Relational Database for the New Millenium. [interaktyvus]. [žiūrėta 2006-02-12]. Prieiga per internetą: www.firebirdsql.org.
4. GNT Lietuva – kompiuterinės technikos tiekėjas. [interaktyvus]. [žiūrėta 2006-02-18]. Prieiga per internetą: <http://www.gnt.lt>.
5. UAB Acme Kompiuterių Komponentai - kompiuterinės technikos tiekėjas. [interaktyvus]. [žiūrėta 2006-02-18]. Prieiga per internetą: <http://www.acc.acme.lt>.
6. UAB “ELKO Kaunas”- kompiuterinės technikos tiekėjas. [interaktyvus]. [žiūrėta 2006-02-18]. Prieiga per internetą: <http://www.elko.lt>.
7. ServiceNet LT- autorizuotas HP, Compaq, IBM, Epson, Samsung, Hyundai, Philips, NEC, Toshiba, Hitachi, Fujitsu-Siemens, Sampo, Belinea, Acer, LG, BenQ, Lexmark įrangos serviso centras. [interaktyvus]. [žiūrėta 2006-02-18]. Prieiga per internetą: <http://www.servicenet.lt>.
8. Tarptautinis projektas SWEBOK Programų inžinerijos žinių vienetams apibūdinti (Guide to the SWEBOK). [interaktyvus]. [žiūrėta 2006-03-10]. Prieiga per internetą: <http://www.swebok.org>.
9. Volere Requirements Specification Template. [interaktyvus]. [žiūrėta 2006-02-07]. Prieiga per internetą: <http://atlsysguild.com/GuildSite/Robs/Template.html>.
10. Apache Web serverio e-puslapis. [interaktyvus]. [žiūrėta 2006-01-22]. Prieiga per internetą: <http://www.apache.org>.
11. WEB serverio teksto preprocesoriaus PHP e-puslapis (lietuviškas). [interaktyvus]. [žiūrėta 2006-11-20]. Prieiga per internetą: <http://www.php.lt>.
12. WEB serverio teksto preprocesoriaus PHP e-puslapis. [interaktyvus]. [žiūrėta 2006-11-23]. Prieiga per internetą: <http://www.php.net>.
13. JANKEVIČIUS, V. Atvirojo kodo duomenų bazių serverių analizė: tarpuniversitetinės konferencijos pranešimų medžiaga. Kaunas, 2006, 139-143 p.

Priedai