

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA

EVALDAS ZIMNICKAS

PRIEIGOS TEISIŲ VALDYMO METODAI IR JŲ NAŠUMO  
TYRIMAS

Magistro darbas

Darbo vadovas  
doc. dr. T. Blažauskas

KAUNAS, 2014

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ INŽINERIJOS KATEDRA

EVALDAS ZIMNICKAS

PRIEIGOS TEISIŲ VALDYMO METODAI IR JŲ NAŠUMO  
TYRIMAS

Magistro darbas

Darbo vadovas:  
doc. dr. T. Blažauskas  
2014-05-

Recenzentas:  
lect. J. Jakutavičius  
2014-05-

Atliko:  
IFM-2/2 gr. studentas  
Evaldas Zimnickas  
2014-05-

KAUNAS, 2014

## AUTENTIŠKUMO PATVIRTINIMAS

### AUTORIŲ GARANTINIS RAŠTAS DĖL PATEIKIAMO KŪRINIO

2014-05-22 d.

Kaunas

**Autorius,** \_\_\_\_\_ *Evaldas Zimnickas*

(vardas, pavardė)

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas (toliau vadinama – Kūrinyje)

*Prieigos teisių valdymo metodai ir jų našumo tyrimas*

(kūrinio pavadinimas)

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

- 1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
- 2) Kūrinyje nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
- 3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
- 4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
- 5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
- 6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
- 7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

**Autorius**

\_\_\_\_\_  
(vardas, pavardė)

\_\_\_\_\_  
(parašas)

# TURINYS

1	TERMINŲ BEI SANTRUMPŲ ŽODYNAS.....	11
2	ĮVADAS .....	12
3	ANALITINĖ DALIS .....	14
3.1	Formalieji informacijos saugos modeliai .....	14
3.1.1	Bell-La Padula .....	14
3.1.2	Biba .....	14
3.2	Prieigos kontrolės metodai .....	15
3.2.1	Rolėmis pagrįsta prieigos kontrolė (RBAC).....	15
3.2.2	Atributais paremta prieigos kontrolė (ABAC).....	16
3.2.3	Atributais ir rolėmis paremtos prieigos kontrolės susiejimas (ABAC + roles) ..	16
3.3	Egzistuojantys vartotojų teisių patikrinimo metodai ASP.NET MVC karkasui. ....	17
3.4	Podėlio panaudojimas optimizuojant skaičiavimus IS sistemose .....	17
3.5	ASP.NET įrankiai sąsajai su duomenų baze .....	19
3.6	Analitinės dalies rezultatai .....	20
4	PROJEKTINĖ DALIS.....	21
4.1	Projekto kūrimo pagrindimas .....	21
4.2	Sistemos paskirtis ir panaudos atvejai.....	21
4.3	Architektūros tikslai ir apribojimai .....	23
4.4	Reikalavimai sistemai.....	24
4.4.1	Funkciniai reikalavimai .....	24
4.4.2	Nefunkciniai reikalavimai.....	25
4.4.2.1	Reikalavimai .....	26
4.5	Teisių valdymo serviso architektūra.....	27
4.5.1	Sistemos statinis vaizdas .....	27
4.5.2	Sistemos dinaminis vaizdas .....	29
5	TYRIMO IR EKSPERIMENTINĖ DALIS .....	31
5.1	Atliekami tyrimai ir metodologija .....	31
5.2	Tiriamosios programinės įrangos realizacija.....	32
5.3	Tyrimui naudojamos įrangos bei parametrų aprašas .....	33
5.4	Eksperimentinių tyrimų rezultatai .....	33
5.4.1	Sistemų atsako laiko priklausomybės nuo prisijungusių vartotojų skaičiaus tyrimas. 33	
5.4.2	Sistemoms reikiamos operatyvinės atminties kiekio priklausomybės nuo prisijungusių vartotojų skaičiaus tyrimas. ....	36

5.4.3	Sistemų atsako laiko priklausomybė nuo turimų funkcijų skaičiaus tyrimas. ....	38
5.4.4	Sistemoms reikiamos operatyvinės atminties kiekio priklausomybės nuo turimų funkcijų skaičiaus tyrimas. ....	40
5.4.5	Sistemų atsako laiko priklausomybė nuo turimų vartotojų grupių skaičiaus tyrimas. ....	43
5.4.6	Sistemoms reikiamos operatyvinės atminties kiekio priklausomybė nuo turimų vartotojų grupių skaičiaus tyrimas. ....	45
6	IŠVADOS .....	48
7	LITERATŪROS SARASAS .....	49
8	PRIEDAI.....	51

## **LENTELIŲ SĄRAŠAS**

1 lentelė. IS dydžių parametrai. ....	33
2 lentelė. Gautų ir numatomų rezultatų skirtumai. ....	45

## PAVEIKSLŲ SĄRAŠAS

1 pav. <i>Bell-La Padula</i> prieiga prie objektų pagal saugumo lygius [1].	14
2 pav. <i>Biba</i> modelio procesų prieiga prie objektų pagal vientisumo lygius [3].	15
3 pav. <i>RBAC</i> prieigos kontrolės metodas [4].	15
4 pav. <i>ABAC</i> prieigos kontrolės metodas [4].	16
5 pav. <i>ABAC</i> prieigos kontrolės metodas [4].	17
6 pav. IS programos sąveika tarp podėlio ir duomenų bazės [10].	18
7 pav. Įvykiai, kurie lemia objekto, išsaugoto podėlyje, galiojimo pabaigą. [9].	18
8 pav. <i>Entity Framework</i> prieigos prie duomenų architektūra. [13].	19
9 pav. LSMU ir LSMUSIS IS kūrimo tikslų sąveika	22
10 pav. LSMUSIS IS sąsajos su išorinėmis sistemomis.	23
11 pav. IS teisių valdymo metodo klasių diagrama	27
12 pav. IS vartotojų teisių grupių administravimo klasių diagrama	28
13 pav. LSMUSIS IS duomenų bazės fragmentas, susijęs su teisių valdymo metodu.	29
14 pav. Vartotojo prieigos prie IS funkcijos veiklos diagrama.	30
15 pav. IS vartotojų teisių grupių redagavimo veiklos diagrama.	30
16 pav. Realizuoto egzistuojančio LSMUSIS IS teisių valdymo metodo klasių diagrama	31
17 pav. Realizuoto stimulatoriaus klasių diagrama.	32
18 pav. Simulatoriaus matavimų procesas.	32
19 pav. Atsako laiko priklausomybė nuo vartotojų skaičiaus, esant S dydžio sistemai.	34
20 pav. Atsako laiko priklausomybė nuo vartotojų skaičiaus, esant M dydžio sistemai.	34
21 pav. Atsako laiko priklausomybė nuo vartotojų skaičiaus, esant L dydžio sistemai.	35
22 pav. Atsako laiko koeficiento priklausomybė nuo vartotojų skaičiaus.	35
23 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų skaičiaus, esant S dydžio sistemai.	36
24 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų skaičiaus, esant M dydžio sistemai.	36
25 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų skaičiaus, esant L dydžio sistemai.	37
26 pav. Užimamos atminties koeficiento priklausomybė nuo vartotojų skaičiaus.	38
24 pav. Atsako laiko priklausomybė nuo funkcijų skaičiaus, esant S dydžio sistemai	38
28 pav. Atsako laiko priklausomybė nuo funkcijų skaičiaus, esant M dydžio sistemai	39
29 pav. Atsako laiko priklausomybė nuo funkcijų skaičiaus, esant L dydžio sistemai	39
30 pav. Atsako laiko koeficiento priklausomybė nuo funkcijų skaičiaus sistemoje.	40
31 pav. Sistemos užimamos atminties priklausomybė nuo funkcijų skaičiaus, esant S dydžio sistemai.	41
32 pav. Sistemos užimamos atminties priklausomybė nuo funkcijų skaičiaus, esant M dydžio sistemai.	41
33 pav. Sistemos užimamos atminties priklausomybė nuo funkcijų skaičiaus, esant L dydžio sistemai.	42
34 pav. Užimamos atminties koeficiento priklausomybė, nuo funkcijų skaičiaus.	42
35 pav. Atsako laiko priklausomybė nuo vartotojų grupių skaičiaus, esant S dydžio sistemai.	43
36 pav. Atsako laiko priklausomybė nuo vartotojų grupių skaičiaus, esant M dydžio sistemai.	43
37 pav. Atsako laiko priklausomybė nuo vartotojų grupių skaičiaus, esant L dydžio sistemai.	44
38 pav. Atsako laiko koeficiento priklausomybė nuo vartotojų grupių skaičiaus sistemoje.	44
39 pav. Sistemos užimamos atminties priklausomybė, nuo vartotojų grupių skaičiaus esant S dydžio sistemai.	45

40 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų grupių skaičiaus, esant M dydžio sistemai. ....	46
41 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų grupių skaičiaus, esant L dydžio sistemai. ....	46
42 pav. Užimamos atminties koeficiento priklausomybė nuo vartotojų grupių skaičiaus. ....	47



# PRIEIGOS TEISIŲ VALDYMO METODAI IR JŲ NAŠUMO TYRIMAS

## *Santrauka*

Kiekviena IS turi funkcijas, kuriomis gali naudotis tik vartotojai, turintys tam teisę. Siekiant užtikrinti tai, reikia priskirti vartotojui prieigos teises ir jas tikrinti kiekvienos užklausoje į serverį metu. Tai turi būti atliekama kuo sparčiau, nes esant dideliame užklausoje kiekiui vienu metu, ši patikra gali užtrukti ilgiau negu yra nurodytas minimalus atsako laikas IS specifikacijoje. Tai itin aktualu šiuolaikinėms IS, nes jose vis plačiau naudojama interaktyvi vartotojo sąsaja, kuri dinamiškai užkrauna duomenis, bet kartu ir padidina užklausoje kiekį į serverį.

Darbo tikslas yra ištirti IS parametrų (vartotojų, teisių grupių ir funkcijų skaičiaus) įtaką sistemos vartotojų teisių valdymo metodo spartai ir pasiūlyti metodą, kuris pasižymėtų našumu ir nereikalautų didelio kiekio operatyvios atminties.

Šiam tikslui įgyvendinti yra pasirinkta Lietuvos sveikatos mokslų universiteto studijų informacinė sistema, kurioje realizuotas pasiūlytas teisių valdymo metodas. Siekiant įrodyti pasiūlyto metodo pranašumą ir ištirti IS parametrų įtaką šio metodo našumui buvo realizuotas simulatorius. Jo pagalba sudaryta reali apkrova IS ir išmatuoti atsako laiko ir užimamos atminties parametrai. Aptarti gauti rezultatai ir pateikiamos išvados.

## *Raktiniai žodžiai*

IS teisių valdymo metodas, vartotojų teisių grupės, podėlio panaudojimas IS, IS funkcijų pasiekiamumas tik autorizuotiems vartotojams, turintiems tam teisę, IS užklausoje atsako laiko mažinimas.

# RESEARCH OF ACCESS RIGHTS MANAGEMENT METHODS AND PERFORMANCE

## *Summary*

Each IS has features that are available only to users with the access right to them. In order to achieve this it is necessary to grant access rights to the user and to filter each request to the server at the time. This has to be done very fast, because at the very huge amount of requests at the same time, filtering can take longer than the specified minimum response time in IS specification. This is especially true in today's IS, as they are increasingly used an interactive user interface, which imposes a dynamic data, but also increases the number of queries to the server.

The aim is to explore the parameters (users, rights groups and the number of features) on the system of user rights management method for speed and propose a method to exhibit performance and requires a large amount of RAM.

This is achieved by the selected Lithuanian University of Health Sciences education information system, which is realized in the proposed rights management method. In order to prove the superiority of the proposed method and to investigate the influence of the parameters of the method performance has been realized the simulator, which simulate a real load of IS and measure response time and amount of used memory. Results are discussed and conclusions are made.

## *Keywords*

IS rights management, users rights groups, using a cache in the IS, IS function available only to authorized users with the access rights. Query to IS response time reduction.

## 1 TERMINŲ BEI SANTRUMPŲ ŽODYNAS

LSMUSIS – Lietuvos sveikatos mokslų universiteto studijų informacinė sistema.

IS – Informacinė sistema.

PI – Programinė įranga

Simulatorius – Programinė įranga, kuri simuliuoja tam tikrą aplinką, veiksmus atitinkančius realybę.

AD (angl. *ActiveDirectory*) – *Microsoft* programinė įranga, skirta vartotojų kompiuterių autorizacijai.

Entity FrameWork – Programinės įrangos karkasas skirtas sąsajai tarp duomenų bazės ir programinės įrangos.

NHibernate – Programinės įrangos karkasas skirtas sąsajai tarp duomenų bazės ir programinės įrangos.

XML - Išplečiama žymų kalba (Extensible markup language).

Microsoft – Korporacija, užsiimanti programinės įrangos kūrimu.

Vartotojas – žmogus, naudojantis sistemą.

IS funkcija – Tam tikras veiksmas, atliekamas IS, kuris turi savo aprašymą.

Aprobavimo aktas – Dokumentas, įrodantis programinės įrangos įdiegimo faktą.

ASP.NET MVC - *Microsoft* programinė įrangos karkasas, paremtas *MVC* šablonu, kuris pritaikytas internetinėms sistemoms.

MVC – Programų architektūrinis modelis (angl. akronimas Model-View-Controller)

IIS (angl. Internet Information Services) - *Microsoft* programinė įranga, skirta internetinių sistemų palaikymui ir administravimui.

SQL (angl. Structured Query Language) - Struktūrizuota užklausų kalba.

Podėlis (angl. **Cache**) - Atmintis, skiriama laikinai padėti programos duomenims, kurių gali vėliau prireikti.

## 2 ĮVADAS

Šis darbas priklauso Programų sistemų inžinerijos studijų programai. Jo paskirtis išsamiai susipažinti su informacinių sistemų vartotojų teisių valdymo metodų našumo tyrimų rezultatais.

### **Darbo problematika ir aktualumas**

Daugelyje veiklos sričių yra naudojama įvairaus pobūdžio informacinė programinė įranga (angl. Information System; toliau – IS). IS tarpusavyje integruojasi, taip sudarydamos didelį ir sudėtingą sistemų tinklą, todėl išauga sistemų sudėtingumas ir atsiranda vartotojų prieigos teisių valdymo problemos.

Kiekviena IS turi funkcijas, kuriomis gali naudotis tik vartotojai turintys tam teisę. Siekiant užtikrinti tai reikia priskirti vartotojui prieigos teises ir jas tikrinti kiekvienos užklausos į serverį metu. Tai turi būti atliekama kuo optimaliau, nes esant dideliame užklausų kiekiui vienu metu, ši patikra gali užtrukti ilgiau negu yra nurodytas minimalus atsako laikas IS specifikacijoje. Tai itin aktualu šiuolaikinėms IS, nes jose vis plačiau naudojama interaktyvi vartotojo sąsaja, kuri dinamiškai užkrauna duomenis, bet kartu ir padidina užklausų kiekį į serverį. Norint, kad sklandžiai veiktų interaktyvi sąsaja, IS turi turėti našų vartotojų teisių prieigos valdymo metodą. Taip pat sistemoms, turinčioms didelį kiekį vartotojų, yra aktualus užimamos operatyvinės atminties kiekis, skirtas vienam vartotojui. Siekiant sutaupyti serverio resursų, reikia kiek įmanoma optimaliau sumažinti išskiriamą atminties kiekį kiekvienam prisijungusiam vartotojui.

### **Darbo tikslas ir uždaviniai**

Darbo tikslas yra ištirti IS parametrų (vartotojų, teisių grupių ir funkcijų skaičiaus) įtaką sistemos vartotojų teisių valdymo metodo spartai ir pasiūlyti metodą, kuris pasižymėtu našumu ir nereikalautu didelio kiekio operatyvios atminties. Šiam tikslui įgyvendinti yra pasirinkta Lietuvos sveikatos mokslų universiteto studijų informacinė sistema (toliau – LSMUSIS) IS. Ji yra įgyvendinama ASP.NET MVC karkase ir integruojama su keletu išorinių sistemų. Taip pat LSMUSIS priklauso universiteto sistemų tinklui, kuriame įgyvendintas vieningas prisijungimas panaudojant Microsoft Active Directory (toliau – AD) servisą. LSMUSIS projekte numatomas nemažas kiekis vartotojų, funkcijų ir teisių grupių. Tai leis patikrinti šių parametrų įtaką teisių valdymo metodo našumui. Taip pat universiteto studijų IS yra saugoma tam tikra dalis konfidencialios studentų informacijos, dėl to turi būti užtikrinamas jos saugumas. Dėl šios priežasties vartotojų teisių prieigos valdymas yra itin aktualus šiai sistemai.

LSMUSIS sistemoje yra naudojamas standartinis vartotojų teisių valdymo metodas, kurio našumas nebetenkina esant didesniai vartotojų aktyvumui. Šiuo darbu bandysime jį pakeisti patobulintu metodu, kuris realizuotas atsižvelgiant į šiuos aspektus:

- Duomenų bazės užklausų optimalumas ir jų kiekis
- Dažnai vykdomų duomenų bazės užklausų ir skaičiavimų rezultatų saugojimas podėlyje
- Prieigos prie funkcijos patikrinimui atliekami skaičiavimai turi būti kuo mažesnės apimties.

Darbo uždaviniai išvedami iš tikslo:

1. Pasiūlyti našų teisių valdymo metodą, kuriame būtų atsižvelgta į aukščiau paminėtus aspektus.
2. Realizuoti pasiūlytą metodą LSMUSIS IS.
3. Realizuoti simulatorių, kuris simuliuotų realias apkrovas ir gebėtų atlikti reikiamus matavimus.
4. Ištirti pasiūlyto metodo spartą ir jam reikalingos operatyvios atminties kiekį.
5. Palyginti senojo ir pasiūlytojo metodo tyrimų rezultatus.

### **Darbo rezultatai ir jų svarba**

Darbo rezultatas yra našus teisių valdymo metodas, kuris yra realizuotas įdiegtoje ir veikiančioje LSMUSIS informacinėje sistemoje. Šio metodo paskirtis - sumažinti IS užtrunkamą laiką vartotojo prieigos prie funkcijos teisėtumo patikrinimui ir naudojamos atminties kiekiui serveryje sumažinimui. Matavimams atlikti realizuotas simulatorius gali būti panaudotas ir kitiems našumo tyrimams atlikti ar simuliuoti realias apkrovas bet kokiai internetu pasiekiamai IS.

### **Darbo struktūra**

**Analitinė dalis (3 skyrius):** Apžvelgiami formalieji informacijos saugos metodai ir išanalizuoti prieigos kontrolės metodai bei įrankiai, kuriuos panaudojus būtų galima pagerinti programinės įrangos našumą.

**Projektinė dalis (4 skyrius):** Specifikuojami reikalavimai teisių valdymo metodui. Pasiūlomas našus teisių valdymo metodas. Aprašomi pasirinkti architektūriniai sprendimai.

**Tyrimo ir eksperimentinė dalis (5 skyrius):** Aprašoma atliekamo tyrimo metodologija ir realizuoto simulatoriaus veikimas. Atlikus tyrimus palyginama realizuoto ir buvusio LSMUSIS IS metodo sparta ir reikiamos operatyvios atminties kiekis.

### 3 ANALITINĖ DALIS

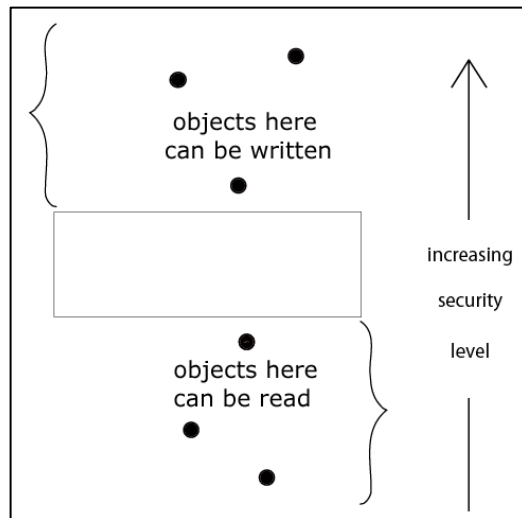
#### 3.1 Formalieji informacijos saugos modeliai

Šiame skyriuje aptariami egzistuojantys informacijos saugos modeliai, trumpai aprašomi kiekvieno iš jų esminiai aspektai.

##### 3.1.1 Bell-La Padula

Bell-La Padula saugumo modelis pateikia konceptualius įrankius, skirtus saugios kompiuterinės sistemos analizei ir projektavimui. Tai vienas iš pirmųjų saugos modelių, kuris pasirodė 1972-1975 metais [**Error! Reference source not found.**]. Jis koncentruotas duomenų konfidencialumui ir kontroliuojamos prieigos prie šios informacijos užtikrinimui. Šiame modelyje įrašai skirstomi į objektus ir subjektus. Visi elementai turi saugumo lygius, kurie nusakomi prieigos taisyklėmis, kurių yra dvi:

- Subjektas negali perskaityti objekto, kuris turi didesnę saugumo lygį.
- Subjektas negali modifikuoti objekto, kai jis turi žemesnę saugumo lygį.



1 pav. *Bell-La Padula* prieiga prie objektų pagal saugumo lygius [1].

Bell-La padula modelyje subjektas gali kurti tik tokio lygio saugumo objektus, kokio lygio yra jis pats. Taip pat subjektai, turintys „Stipriosios žvaigždės“ nustatymą, gali skaityti ir rašyti tik į tokio pat saugumo lygio objektus. Šį saugumo modelį naudoja visos privalomos prieigos kontrolės (angl. mandatory access controls; toliau MAC) sistemos.

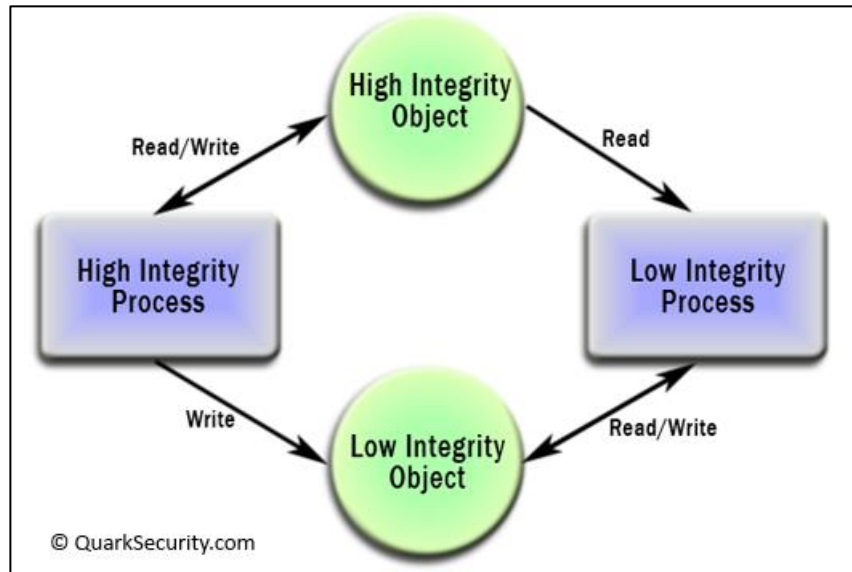
##### 3.1.2 Biba

Priešingai nei Bell-La Padula, šis modelis akcentuoja duomenų vientisumą ir vietoje saugumo lygių yra vientisumo (integrity) lygiai [2]. Šis modelis turi tris griežtas savybes:

- Paprastas vientisumas:  $I(S) \geq I(O)$  - modifikuoti
- Vientisumo apribojimas:  $I(S) \leq I(O)$  - skaityti
- Reikalavimo savybė:  $I(S_1) \geq I(S_2)$  – iškviesti

Remiantis Biba procesų prieigos prie objektų modeliu (2 pav.), galime aprašyti keturias taisykles:

- Aukštą lygį turintis procesas gali skaityti ir rašyti aukštą lygį turintį objektą.
- Aukštą lygį turintis procesas gali tik rašyti žemą lygį turintį objektą.
- Žemą lygį turintis procesas gali skaityti ir rašyti žemą lygį turintį objektą.
- Žemą lygį turintis procesas gali tik skaityti aukštą lygį turintį objektą.

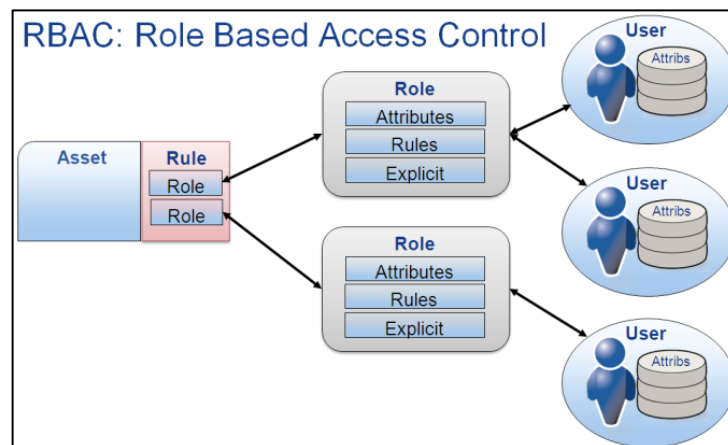


2 pav. Biba modelio procesų prieiga prie objektų pagal vientisumo lygius [3].

## 3.2 Prieigos kontrolės metodai

### 3.2.1 Rolėmis pagrįsta prieigos kontrolė (RBAC)

Rolėmis pagrįstos prieigos kontrolės (angl.. Role-based access control (RBAC); toliau – RBAC) idėja yra ta, kad prieigos teisės prie IS funkcijos yra suteikiamos rolėms, kurios yra priskiriamos vartotojams (3 pav.).

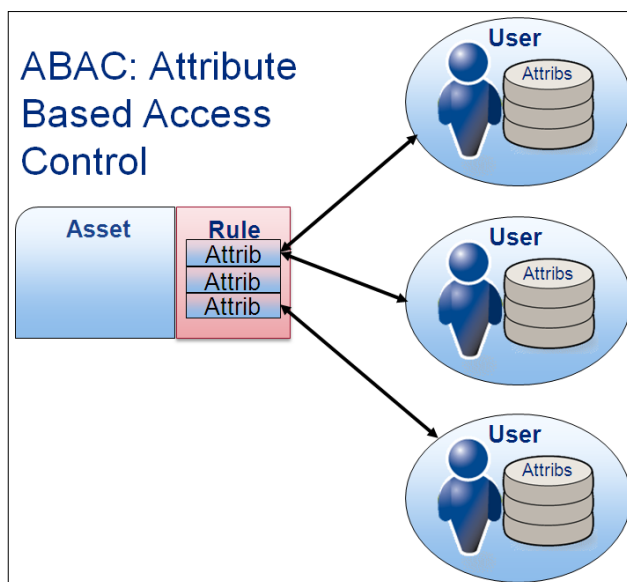


3 pav. RBAC prieigos kontrolės metodas [4].

Taip pat egzistuoja tam tikros taisyklės, kuriomis remiantis yra kontroliuojama prieiga prie IS funkcijų ir duomenų. Taikant šias taisykles vartotojui pagal jų turimus atributus yra priskiriamos rolės. Naudojantis šiuo metodu, esant nemažam skaičiui funkcijų, rolių ir duomenų IS sistemos administravimas tampa komplikuoatas. Norint pridėti vartotojui tam tikrą IS funkciją, reikia surasti ją turinčią rolę ir įsitikinti, kad priskyrus šios rolės funkciją vartotojas nepasieks jam negalimų duomenų. Neradus atitinkamos rolės, tenka kurti naują, o jos pridėjimas dar labiau apsunkina tolesnį IS administravimą.

### 3.2.2 Atributais paremta prieigos kontrolė (ABAC)

Atributais pagrįstos prieigos kontrolės (angl.. Attribute-based access control (ABAC); toliau – ABAC) metodas yra daug paprastesnis nei *RBAC*. Jame nebėra rolių, kurios atlikdavo tarpininko vaidmenį, atsakingą už prieigos kontrolę. Šiame metode kiekviena funkcija ar duomenys turi taisykles, pagal kurias nustatoma, ar vartotojas su tam tikrais atributais turi galimybę prieiti prie šių duomenų (4 pav.).



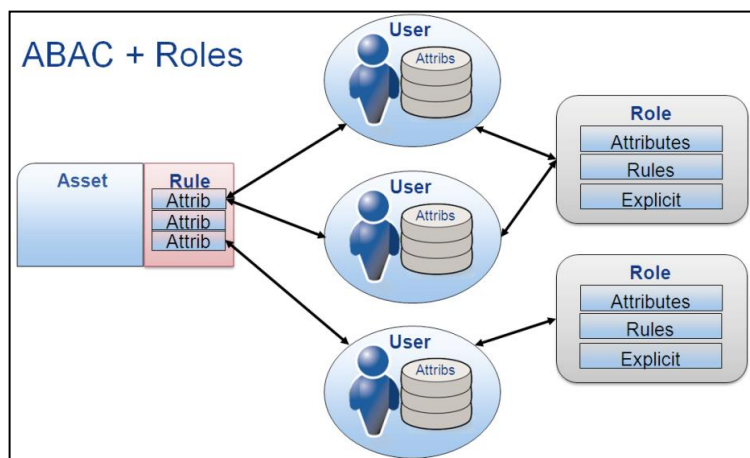
4 pav. ABAC prieigos kontrolės metodas [4].

Šio metodo trūkumas, kad esant nemažam kiekiui vartotojų, užtrunka daug laiko administruojant IS prieigos teises. Tai atsitinka dėl to, kad norint tam tikrai grupei vartotojų suteikti ar atimti tam tikras prieigos teises, tenka modifikuoti kiekvieno vartotojo atributus individualiai.

### 3.2.3 Atributais ir rolėmis paremtos prieigos kontrolės susiejimas (ABAC + roles)

Tai yra patobulintas *ABAC* metodas. Siekiant panaikinti pagrindinį metodo trūkumą jame atsiranda rolės. Tik kitaip nei *RBAC* metode, rolės yra nebe tarpininkas tarp vartotojo ir funkcijos, o siejasi tik su vartotojais, o vartotojo ir funkcijos sąryšis išlieka toks pat kaip *ABAC* (5 pav.).





5 pav. ABAC prieigos kontrolės metodas [4].

Apjungus šių dviejų metodų geriausias savybes gaunamas lengvai administruojamas ir lankstus vartotojų prieigos teisių valdymo metodas. Egzistuoja nemažai *RBAC* ir *ABAC* metodų sujungimų modifikacijų. Daugelyje atvejų projektuojant IS sistemą yra panaudojamas vienas iš aprašytų teisių valdymo metodas, jį pritaikant kuriamai sistemai.

### 3.3 Egzistuojantys vartotojų teisių patikrinimo metodai ASP.NET MVC karkasui.

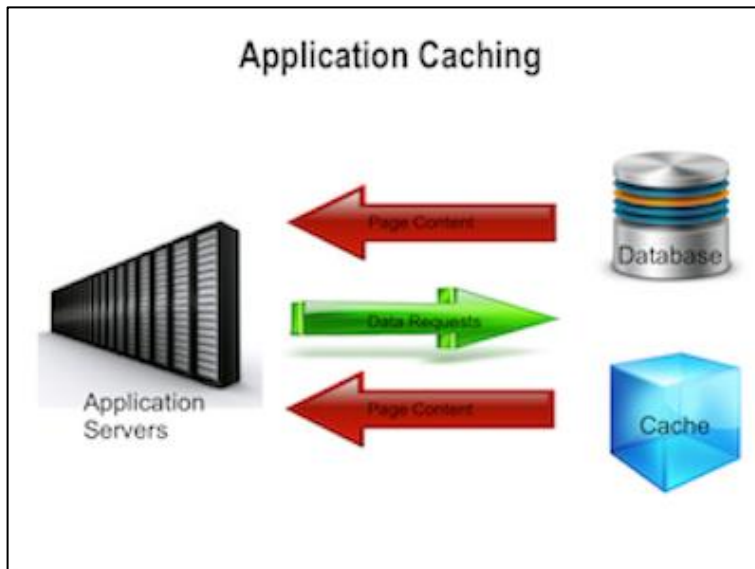
Naudojant ASP.NET MVC karkasą, visa vartotojo sąsaja yra paremta MVC šablonu, tad prieiga prie funkcijų yra per valdiklio (controller) metodus. Tam, kad užtikrintų vartotojui prieigą tik prie jam priskirtų funkcijų, reikia prieš iškviečiant valdiklio metodą patikrinti, ar vartotojas turi teisę jį vykdyti. Daugeliu atveju tai atliekama vienu iš šių dviejų metodų:

- Vartotojui siunčiant užklausą į serverį yra patikrinamas užklausos kelias (route), tai atitinka tam tikrą funkciją sistemoje, ir sutapatinamas su vartotojui priskirtu keliu prie funkcijos. Jei nerandama atitikmens, grąžinama klaida ir vartotojo užklausa blokuojama. Šis metodas yra daug lankstesnis, o pagrindinis šio metodo privalumas jog nereikia aprašinėti prieš kiekvieną funkciją antraštės [11].
- Virš kiekvieno metodo yra aprašoma antraštė (annotation), kokie vartotojai ar jų grupė gali vykdyti šį metodą. Užklausos metu vartotojo prieigos prie funkcijos patikrinimas yra vykdomas tik prieš iškviečiant patį metodą. Jei vartotojas neatitinka antraštėje aprašytų parametrų, grąžinama klaida ir vartotojo užklausa blokuojama. Pagrindinis šio metodo privalumas yra, kad filtruojamos tik tos užklausos, kurios kreipiasi į teisių reikalaujančius metodus, tad mažiau apkrauna sistemą [12].

### 3.4 Podėlio panaudojimas optimizuojant skaičiavimus IS sistemose

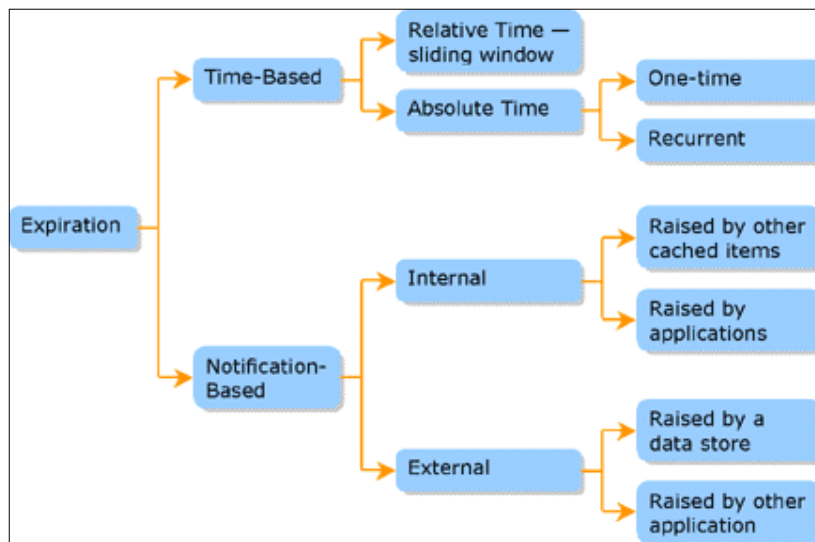
Podėlio panaudojimas optimizuojant IS sistemų skaičiavimus yra labai plačiai naudojamas. Atlikus tam tikrus skaičiavimus, gautus rezultatus, kurių gali dažnai prireikti ateityje, pravartu išsaugoti podėlyje. Saugomi duomenys gali būti privatūs (tik konkretaus vartotojo) arba bendri (prieinami visiems). Prireikus atlikti identiškus skaičiavimus dar kartą,

neberekia skirti laiko, nes jau turime gautą rezultatą, belieka tik išrinkti jį iš podėlio. Taip pat podėlį galima panaudoti ir ilgai trunkančioms duomenų bazės užklausoms, darant prielaidą, kad rezultatas artimiausiu metu nepasikeis. Tokiu atveju sistema, kiekvieną kartą prieš užklaudama duomenų bazę, patikrina, ar nėra norimo rezultato atmintinėje, o neradus jo, gautą rezultatą iš duomenų bazės išsaugoja kitam kartui (6 pav.).



6 pav. IS programos sąveika tarp podėlio ir duomenų bazės [10].

Duomenys podėlyje turi galiojimo terminą, kuris gali būti begalinis arba baigtinis. Trukmės dydį lemia sistemoje egzistuojančios taisyklės (7 pav.).



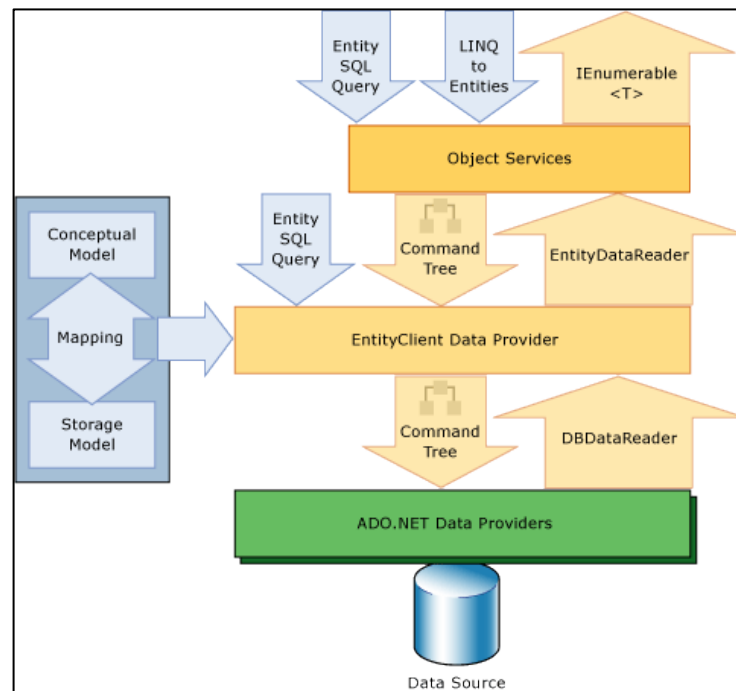
7 pav. Įvykiai, kurie lemia objekto, išsaugoto podėlyje, galiojimo pabaigą. [9]

Tam tikrus duomenis verta išsaugoti podėlyje tik startavus sistemai, o juos atnaujinti, atlikus tik tam tikrą funkciją. Tai IS sistemoje galėtų būti daugiakalbystės tekstai, egzistuojančios vartotojų grupės ir t.t. Panaudojant podėlį moderniose IS, galima ženkliai padidinti atsako laiką. [10].

### 3.5 ASP.NET įrankiai sąsajai su duomenų baze

Sąsajai tarp sistemos ir duomenų bazės yra naudojami tarpininkavimo įrankiai, palengvinantys programavimo darbus, užtikrinantys saugumą ir gebantys optimizuoti užklausas, parašytas LINQ išraiška. Vieni iš populiariausių ir turintys visas aukščiau paminėtas funkcijas yra: Entity Framework ir NHibernate.

Entity Framework (EF) – Tai šiuo metu itin populiarėjantis įrankis, kuris yra palaikomas Microsoft korporacijos. Duomenų išrinkimui iš duomenų bazės yra rašoma LINQ išraiška, iš kurios yra generuojama SQL užklausa. Duomenų bazės lentelės atitinka objektą sistemoje. Šis įrankis geba generuoti klases su ryšiais iš egzistuojančios duomenų bazės ir atvirkščiai. EF yra lankstus ir lengvai konfigūruojamas, taip pat geba saugoti užklausų rezultatus podėlyje (Caching), ženkliai paspartindamas rezultatų gavimą iš duomenų bazės [14].



8 pav. Entity Framework prieigos prie duomenų architektūra. [13]

NHibernate – tai yra gerokai senesnis įrankis lyginant su Entity Framework, bet turi beveik tokį patį funkcionalumą. Keletas esminių skirtumų: [14]

- Visa konfigūracija XML faile.
- Yra daug lankstesnis nei EF.
- Oficialiai nepalaikomas Microsoft.
- Susiejimas su kitais Microsoft įrankiais menkas ar išvis nėra.
- Palyginus su EF, trūksta mokomosios literatūros pradedantiems.

### 3.6 Analitinės dalies rezultatai

Atlikus analizės etapą priimti sprendimai:

- Atsižvelgus į administravimo paprastumą ir tinkamumą LSMUSIS IS, pasirinktas naudoti patobulintas *ABAC* prieigos teisių valdymo metodas.
- Atlikus ASP.NET MVC aplinkoje egzistuojančių teisių valdymo metodų analizę, nuspręsta prieigą prie funkcijų valdyti panaudojant veiksmų aprašus. Kiekviena funkcija IS turės savo teisių rolę, kuri priklausys tam tikrai vartotojų grupei.
- Siekiant, kad patobulintas teisių valdymo metodas būtų našesnis nei senasis, išanalizuoti pagrindiniai spartinimą galintys pagerinti įrankiai.
- Atlikta egzistuojančių įrankių, skirtų saugiai .NET sistemų sąsajai su duomenų bazėmis, analizė. Pasirinkimą labiausiai lėmė Microsoft produkto palaikymas, tad pasirinktas Entity Framework įrankis, nes atsiradus saugumo spragoms jis bus atnaujintas.

## 4 PROJEKTINĖ DALIS

### 4.1 Projekto kūrimo pagrindimas

Lietuvos sveikatos mokslų universiteto vienos iš pagrindinių IS (LSMUSIS) atsako laikas nebetenkina galutinių vartotojų poreikių ir reikalavimų specifikacijoje pateikto minimalaus atsako laiko. Tai lemia nuolatos didėjantis vartotojų skaičius ir besiplečiantis LSMUSIS sistemos funkcionalumas. Tad didžiąją atsako laiko dalį trunka vartotojo prieigos prie funkcijos teisėtumo tikrinimas.

Kadangi IS yra sudėtinga, apima nemažai funkcijų ir neturi analogų, kurie galėtų visiškai pakeisti sistemą, buvo nuspręsta perdaryti arba patobulinti sistemos teisių valdymo metodą, kad jis atitiktų visus specifikacijoje minimus reikalavimus.

Atlikus egzistuojančių metodų analizę, nebuvo rastas nei vienas visiškai tinkantis. Tad buvo nuspręsta perkurti egzistuojantį IS teisių valdymo metodą, panaudojant skaičiavimus spartinančius įrankius ir metodikas.

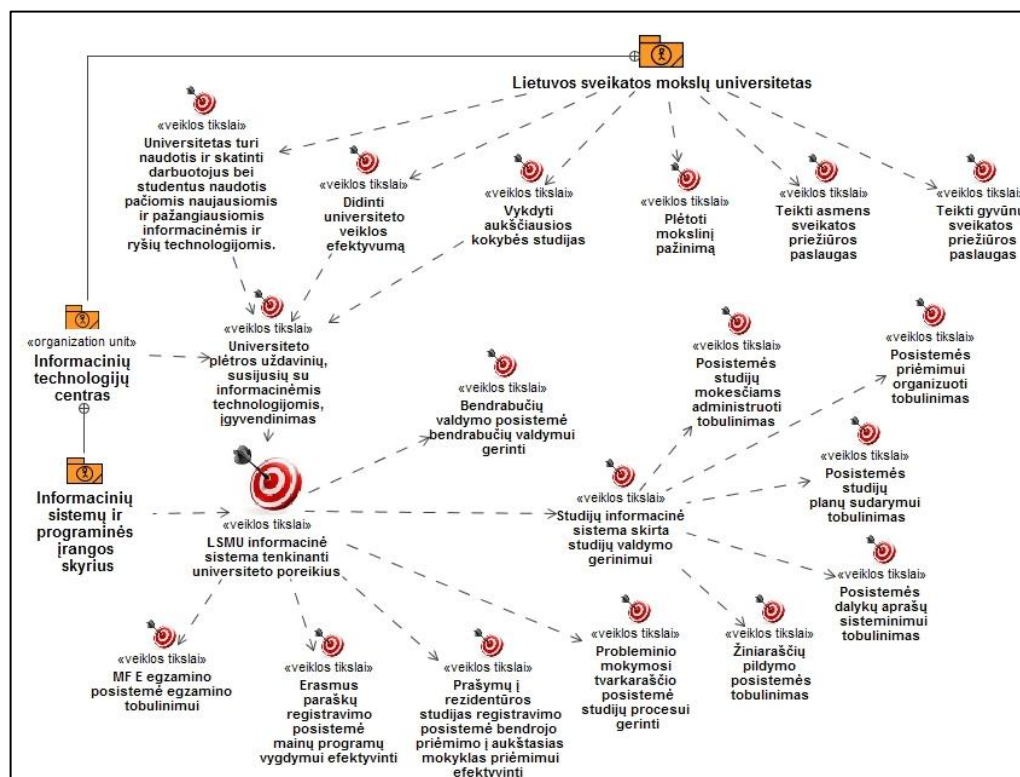
### 4.2 Sistemos paskirtis ir panaudos atvejai

LSMUSIS IS paskirtis - tvarkyti studentų priėmimą, jų studijas, apgyvendinimą, mokesčių apskaitą, organizuoti studijų procesą, tvarkyti studijų dalykų / modulių aprašus, pateikti studijų rezultatus, vykdyti studijų kokybės vertinimo funkcijas ir kitus su studijomis susijusius procesus. Taip pat ši sistema turi pilnai integruotis su kitomis universiteto sistemomis.

Projekto tikslas yra patobulinti LSMUSIS IS teisių valdymo metodą, kuris atitiktų visus keliamus reikalavimus ir padėtų pasiekti IS tikslus, kurie kyla iš universiteto pagrindinių tikslų, t.y., universitetas turi naudotis ir skatinti darbuotojus bei studentus naudotis pačiomis naujausiomis ir pažangiausiomis informacinėmis ir ryšių technologijomis, didinti universiteto veiklos efektyvumą, vykdyti aukščiausios kokybės studijas [5]. Svarbiausi sistemos tikslai yra:

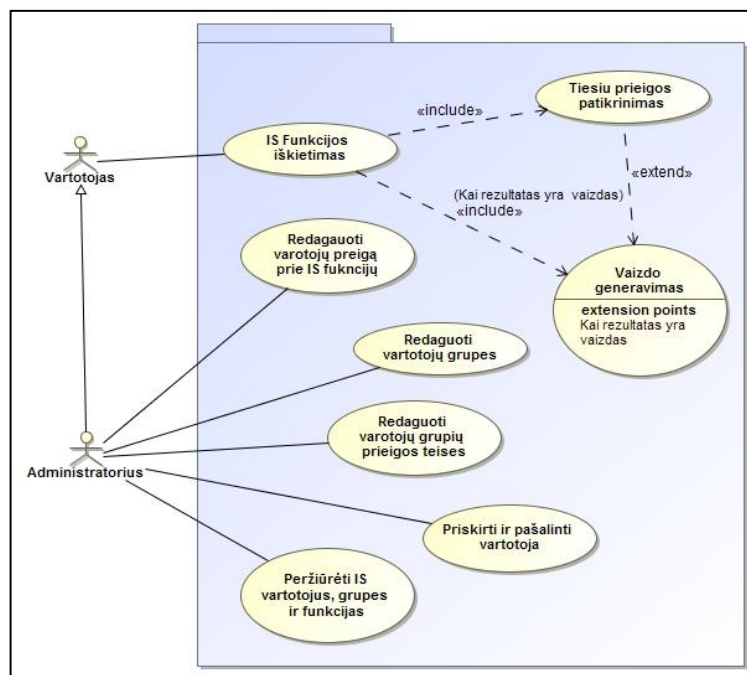
- vykdyti studentų priėmimo procedūras;
- administruoti universiteto bendrabučių teikiamas apgyvendinimo paslaugas;
- modernizuoti universiteto studijų programų dalykų aprašus remiantis Bolonijos procesu;
- organizuoti studijų procesą;
- operatyviai teikti informaciją Švietimo ir mokslo ministerijai;
- vesti apmokėjimo už studijas apskaitą;
- informuoti studentus apie įsipareigojimus, pasiekimus, studijas;
- anketų pagalba vykdyti studijų kokybės vertinimą;
- kaupti, saugoti ir paruošti duomenis diplomų ir jų priedėlių spausdinimui.
- visiškai integruotis su kitomis universitete naudojamomis IS sistemomis;

Tikslai, kurių reikia siekti, tobulinat LSMUSIS IS pavaizduoti 9 paveiksle. Sukurta sistema turi tenkinti universiteto poreikius.



9 pav. LSMU ir LSMUSIS IS kūrimo tikslų sąveika

Paveiksle nr. 10 pateikiama panaudos atvejų diagrama, kurioje matyti metodo teikiamos funkcijos IS administratoriams ir jos vartotojams.



10 pav. Teisių valdymo metodo panaudos atvejų diagrama

Teisių valdymo metodas leis atlikti šias funkcijas:

- Redaguoti kiekvieno vartotojo prieigą prie IS funkcijų.

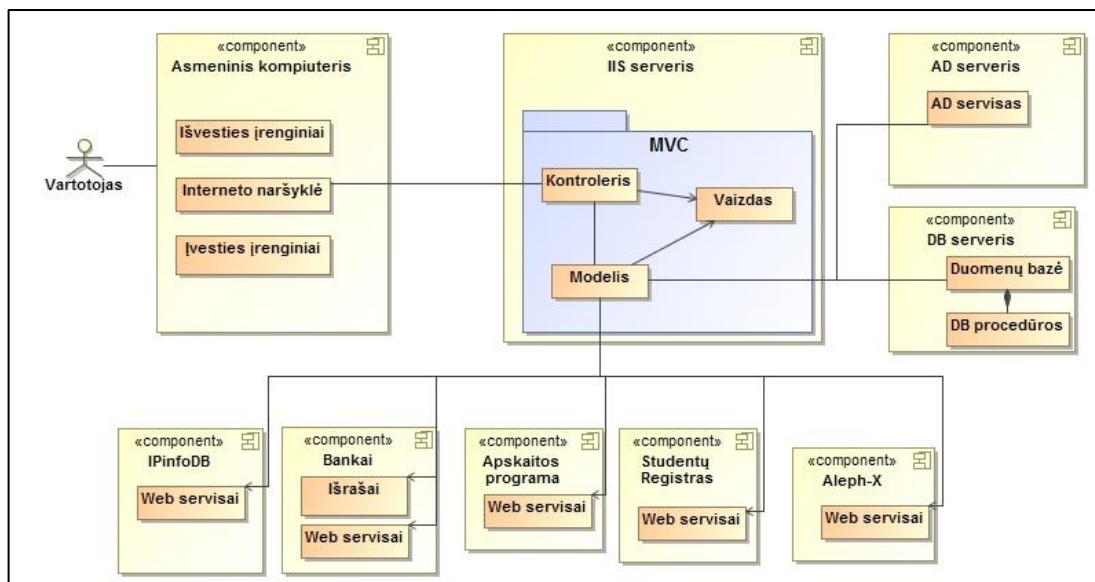
- Suskirstyti IS vartotojus į grupes.
- Redaguoti vartotojų grupėms priskirtas IS funkcijas.
- Filtruoti vartotojų prieigą prie funkcijos, pagal jam priskirtas teises.
- Sudarinėjant vartotojui matomą vaizdą, patikrinti, kuriuos mygtukus, laukus ar kitus grafinės sąsajos komponentus gali matyti vartotojas.

#### 4.3 Architektūros tikslai ir apribojimai

Kuriamo metodo architektūriniai tikslai yra:

- pakartotinis klasių panaudojimas
- nusakančių paskirtį vardų naudojimas
- komponentais paremta architektūra
- egzistuojančių įrankių panaudojimas

LSMUSIS sistema realizuota panaudojant ASP.NET MVC karkasą. Kuriamas teisių valdymo metodas turi visiškai integruotis į IS ir apimti visas jos funkcijas ir kontroliuoti vartotojų prieigą per išorines sistemas. LSMUSIS sąsajos su išorinėmis sistemomis pavaizduotos 11 paveiksle.



10 pav. LSMUSIS IS sąsajos su išorinėmis sistemomis

Technologinė ir fizinė sistemos diegimo aplinka:

- Sistema veikia Universiteto serverio virtualioje mašinoje, Microsoft Windows Server 2008 R2 operacinėje aplinkoje.
- Duomenų bazė realizuota Microsoft SQL Server 2008 R2 duomenų bazėje.
- Sistemai palaikyti naudojama IIS 7,5 puslapių valdymo paslauga.
- Sistemos vartotojų identifikavimui naudojama Microsoft AD servisas.
- LSMUSIS IS realizuota ASP.NET MVC 4 karkase.

Dėl didelio vartotojų kiekio, dėl skirtingai atliekamų funkcijų pasirinkta internetinė informacinės sistemos įgyvendinimo architektūra. Sistema diegiama serveryje, kuriame palaikoma IIS7,5 paslauga, į kurią kreipiasi vartotojas interneto naršyklės pagalba.

Virtualioje mašinoje yra išskirti serveriai duomenų saugojimui, vartotojų autentifikavimui, internetinės sistemos puslapių realizavimui.

Vaizdas interneto puslapiuose atvaizduojamas naudojantis HTML, JavaScript, CSS bei panaudojant jau sukurtas elementų bibliotekas.

#### 4.4 Reikalavimai sistemai

Iš detalizuotų LSMUSIS IS panaudos atvejų, buvo atrinkti reikalavimai susiję su teisių valdymo metodu. Daugelis šių funkcinų reikalavimų susiję su vartotojų prieigos prie IS administravimu. Nefunkciniai reikalavimai apibrėžia reikalavimus sistemos saugumą ir vykdymo charakteristikas. Svarbiausi iš atrinktų reikalavimų yra šie:

##### 4.4.1 Funkciniai reikalavimai

<u>Reikalavimas#:</u>	13	<u>Reikalavimo tipas:</u>	9	<u>Ivykis/panaudojimo atvejis#:</u>	22
<u>Aprašymas:</u>	IS administratorius turi turėti galimybę administruoti vartotojų teisių grupes.				
<u>Pagrindimas:</u>	Administratorius galės pridėti ar pašalinti vartotojui priklausančias grupes.				
<u>Šaltinis:</u>	Veiklos taisyklės				
<u>Tikimo kriterijus:</u>	Administratoriams galima bus redaguoti vartotojui priklausančias teisių grupes.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybė</u>	14	<u>Konfliktai:</u>			
<u>Papildoma medžiaga:</u>					
<u>Istorija</u>	2013-10-04				

<u>Reikalavimas#:</u>	14	<u>Reikalavimo tipas:</u>	9	<u>Ivykis/panaudojimo atvejis#:</u>	23
<u>Aprašymas:</u>	IS administratorius turi turėti galimybę administruoti vartotojų teisių grupes.				
<u>Pagrindimas:</u>	Administratoriams bus galima redaguoti teisių grupes, pridėti ar pašalinti joms priklausančias funkcijas bei matyti IS egzistuojančių grupių sąrašą.				
<u>Šaltinis:</u>	Veiklos taisyklės				
<u>Tikimo kriterijus:</u>	Administratoriams galima bus redaguoti vartotojų teisių grupes.				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybė</u>		<u>Konfliktai:</u>			
<u>Papildoma medžiaga:</u>					
<u>Istorija</u>	2013-10-04				

<u>Reikalavimas#:</u>	15	<u>Reikalavimo tipas:</u>	9	<u>Ivykis/panaudojimo atvejis#:</u>	24
<u>Aprašymas:</u>	Vartotojų slaptažodžio keitimas				
<u>Pagrindimas:</u>	Vartotojas keičia slaptažodį. Slaptažodį gali žinoti tik pats vartotojas.				
<u>Šaltinis:</u>	Vartotojas ir AD				
<u>Tikimo kriterijus:</u>	Pagal nutylėjimą AD sukurtas slaptažodis keičiamas į vartotojo įvestą ir patvirtintą slaptažodį				
<u>Užsakovo tenkinimas:</u>	5	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybė</u>	17	<u>Konfliktai:</u>			



<u>Papildoma medžiaga:</u>	Universiteto asmens duomenų apsaugos tvarkos įsakymas		
<u>Istorija</u>	2013-11-04		

<u>Reikalavimas#:</u>	19	<u>Reikalavimo tipas:</u>	9	<u>Ivykis/panaudojimo atvejis#:</u>	24
<u>Aprašymas:</u>	IS Vartotojų sąrašas				
<u>Pagrindimas:</u>	Visų IS egzistuojančių vartotojų sąrašas, kuriame matomi prisijungimo vardai, paskutinė prisijungimo data ir pagrindinė vartotojo informacija.				
<u>Šaltinis:</u>					
<u>Tikimo kriterijus:</u>	Galima administruoti visus IS esančių vartotojų duomenis.				
<u>Užsakovo tenkinimas:</u>	2	<u>Užsakovo netenkinimas:</u>	5		
<u>Priklausomybė</u>			<u>Konfliktai:</u>		
<u>Papildoma medžiaga:</u>					
<u>Istorija</u>	2013-11-04				

#### 4.4.2 Nefunkciniai reikalavimai

<u>Reikalavimas#:</u>	17	<u>Reikalavimo tipas:</u>	15	<u>Ivykis/panaudojimo atvejis#:</u>	24
<u>Aprašymas:</u>	Vartotojų slaptažodžiai tenkina asmens duomenų apsaugos tvarką				
<u>Pagrindimas:</u>	Slaptažodis ne per trumpas ir nėra paprastas žodis				
<u>Šaltinis:</u>	Microsoft AD				
<u>Tikimo kriterijus:</u>	Sudarytas bent iš 7 raidžių ir skaitmenų				
<u>Užsakovo tenkinimas:</u>	3	<u>Užsakovo netenkinimas:</u>	1		
<u>Priklausomybė</u>			<u>Konfliktai:</u>		
<u>Papildoma medžiaga:</u>	Universiteto asmens duomenų apsaugos tvarkos įsakymas				
<u>Istorija</u>	2013-11-07				

<u>Reikalavimas#:</u>	26	<u>Reikalavimo tipas:</u>	15	<u>Ivykis/panaudojimo atvejis#:</u>	26
<u>Aprašymas:</u>	Vartotojų prieiga tik prie jiems priskirtų funkcijų ir duomenų				
<u>Pagrindimas:</u>	Visi IS vartotojai negali prieiti prie jiems draudžiamų ar nepriskirtų duomenų ir funkcijų.				
<u>Šaltinis:</u>	Veiklos taisyklės				
<u>Tikimo kriterijus:</u>	Prieš vykdant kiekvieną funkciją, patikrinamas vartotojo prieigos prie jos teisėtumas.				
<u>Užsakovo tenkinimas:</u>	3	<u>Užsakovo netenkinimas:</u>	1		
<u>Priklausomybė</u>			<u>Konfliktai:</u>		
<u>Papildoma medžiaga:</u>					
<u>Istorija</u>	2013-11-08				

#### 4.4.2.1 Reikalavimai vykdymo charakteristikoms (Performance)

IS pasižymės tokiais vykdymo charakteristikomis:

- Ilgesnės kaip 10 sekundžių skaičiavimo laiko reikalaujančios procedūros bus atliekamos nustatytu paros metu arba apie jų vykdymo trukmę bus pranešama iš anksto;
- Priklausomai nuo kompiuterio pajėgumo, minimalus puslapių krovimo laikas – 3 sekundės;
- Vienu metu gali būti prisijungę 1000 vartotojų;
- Vienu metu gali būti vykdoma iki 1000 užklausimų;
- Sistemoje mažiausia dalis yra šimtoji vieneto dalis;
- Sistemoje reikšmių diapazonas gali varijuoti:
  - Žodinėms reikšmėms nuo 1 iki 800 simbolių;
  - Skaitinėms reikšmėms nuo 0 iki 1 000 000 000 000;
  - Datos ir laiko nuo 1800 m. sausio mėn. 1 d. 00:00:00:00 iki 2100 m. gruodžio mėn. 31 d. 23:59:59:99;
  - Paveikslams iki 1 MB;
- Duomenų bazė turi talpinti nuo 3 GB iki 10 GB duomenų, su galimybe didinti, esant poreikiui;
- Sistema turi veikti 94 proc. paros laiko;
- Turi būti saugomos duomenų kopijos;
- Pakeitimai turi būti atsekami;
- Vykdomas įvedamų duomenų sutikrinimas;

#### 4.4.2.2 Reikalavimai saugumui (Security)

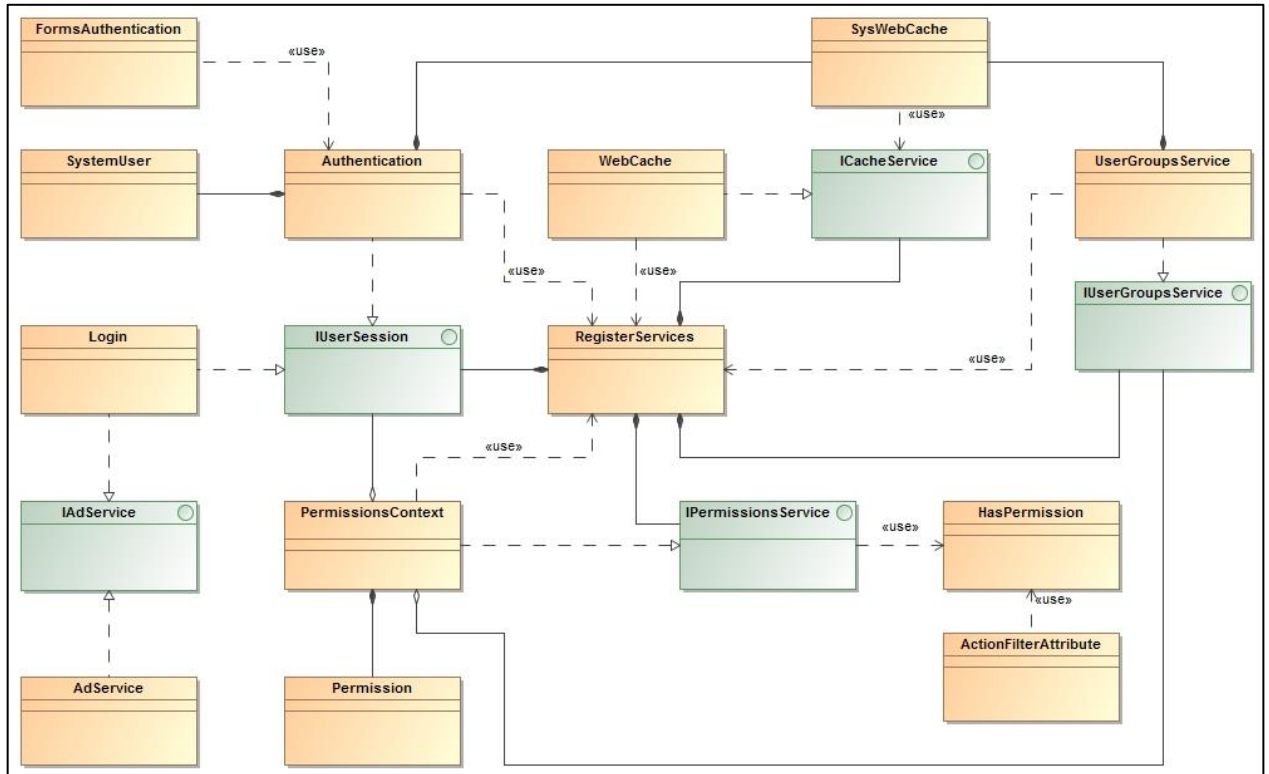
Išskiriami pagrindiniai IS saugumo aspektai:

- Konfidencialumas. Sistemoje turi būti garantuotas asmens duomenų saugumas. Naudotis IS gali tik identifikuoti ir autorizuoti asmenys. Vartotojų slaptažodžiai turi atitikti universiteto dokumentuose numatytus reikalavimus – bent 7 simboliai, iš kurių nors 1 skaitmuo ir 1 raidė. Viešai bus pateikiami tik patvirtinti programų ir dalykų/modulių aprašai ir tam skirta informacija.
- Vientisumas. Sistemoje vedami duomenys saugomi su autoriaus ir laiko informacija. Patvirtinti duomenys tampa neredaguotini. Tik patvirtinti finansiniai, studijų, įvertinimų duomenys pasiekia kitą veiklos etapą ir veiklos aktorių.
- Pasiekiamumas. Duomenys per fiksuotą laiką turi būti pasiekiami autorizuotiems asmenims. Gedimo ar piktybinio sunaikinimo atveju turi būti atstatomi paros senumo duomenys. Per 48 valandas nuo sistemos sustojimo turi būti atstatyta jos veikla.

## 4.5 Teisių valdymo serviso architektūra

### 4.5.1 Sistemos statinis vaizdas

IS teisių valdymo komponente yra panaudoti .NET karkaso turimi įrankiai. Bendra metodo realizacijos klasių diagrama pavaizduota 11 paveiksle.



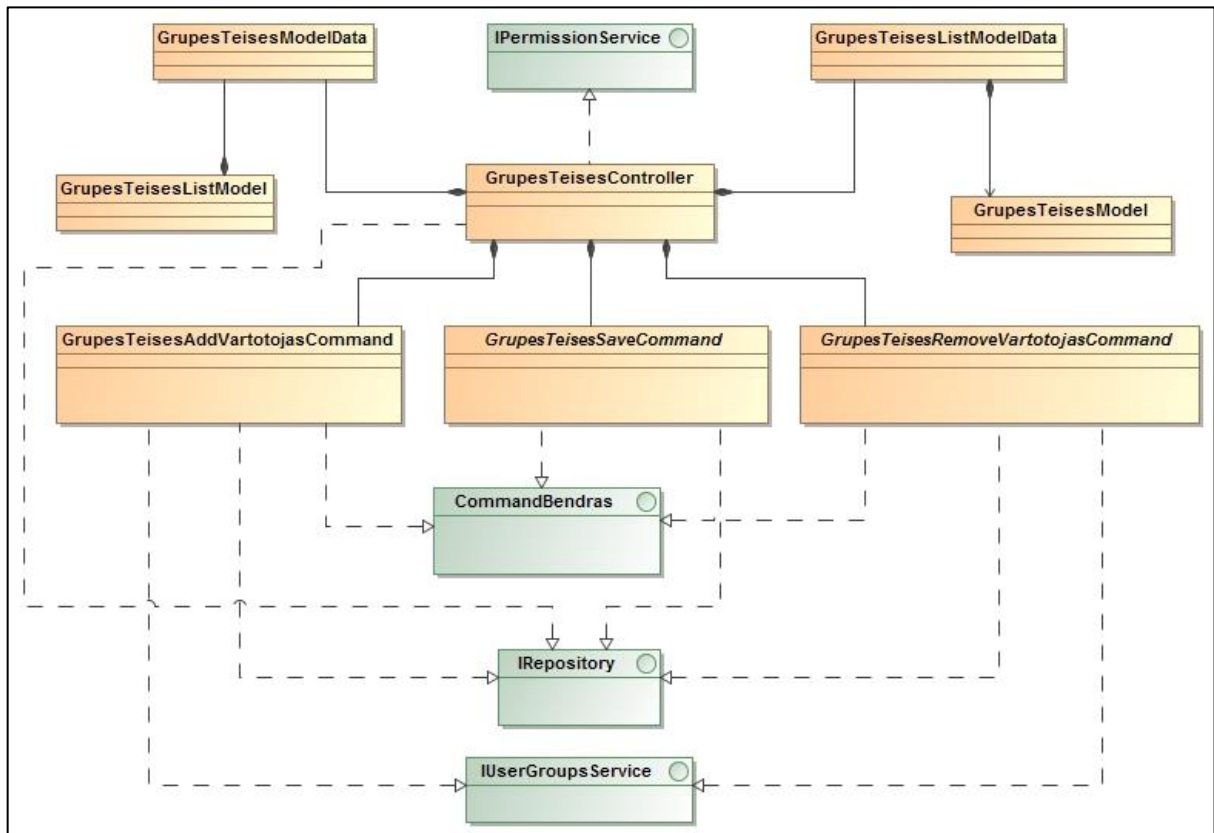
11 pav. IS teisių valdymo metodo klasių diagrama

Realizuotų klasių aprašymai ir paskirtis:

- *FormsAuthentication* – ASP.NET karkaso klasė, kurios pagalba yra suteikiama vartotojui prisijungimo sesija.
- *SystemUser* – klasė, kurioje saugoma prisijungusio vartotojo informacija: asmens duomenys, priskirtos teisių grupės, identifikavimo duomenys.
- *Login* – klasė atsakinga, už vartotojo prisijungimą prie IS per AD sistemą.
- *AdService* – klasė, kurioje realizuota saugi sąsaja tarp IS ir AD sistemos. Tam panaudota .NET karkaso biblioteka [6].
- *Authentication* – klasė, kurioje realizuota visos su vartotojo sesija susijusios funkcijos. Ji pagal AD sistemos pateiktą prisijungimą esamai vartotojo sesijai suteikia IS jam priskirtas teisių grupes. Taip pat joje realizuotos sąlyginės teisių grupės (teisių grupės kurios automatiškai priskiriamos pagal tam tikrus vartotojo atributus, pvz. Studentas).
- *PermissionsContext* – klasė, atsakinga už IS teisių prieigos valdymą. Pagal vartotojo turimus teisių atributus yra patikrinamas jo prieigos prie funkcijos galimumas.
- *Permission* – klasė, kuri saugo duomenis apie sistemos funkciją.

- *WebCache* – klasė atsakinga už sistemos bendrus saugomus duomenis podėlyje.
- *SysWebCache* - klasė atsakinga už IS vartotojo saugomus duomenis podėlyje.
- *RegisterServices* – klasė, kuri paleidžia visus servisus startavus IS. Tam panaudota *SructureMap* biblioteka [7].
- *HasPermission* – Tai klasė, patikrina gražina ar vartotojas turi teisę prie funkcijos, pagal jos kodą.
- *ActionFilterAttribute* – ASP.NET aprašo praplėtimas, kuris iškviečia teisių valdymo metodą prieš vykdant IS funkciją.
- *UserGroupService* – klasė, kuri atsakinga už vartotojų teisių grupių saugojimą podėlyje ir jų išrinkimą kiekvienam vartotojui.

IS vartotojų teisių grupių administravimas yra realizuotas kaip ir visos kitos LSMUSIS sistemos funkcijos. Grupių administravimo klasių diagrama pateikiama 12 paveiksle.



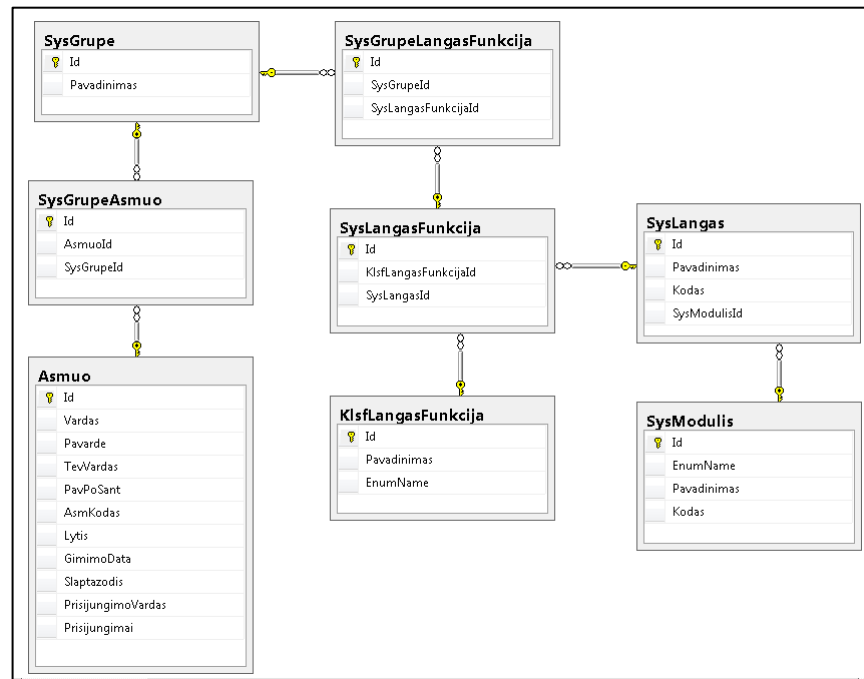
12 pav. IS vartotojų teisių grupių administravimo klasių diagrama

Realizuotų klasių aprašymai ir paskirtis:

- *GrupesTeisesController* – ASP.NET MVC karkaso elementas, kuriame aprašytos visos teisių grupių funkcijos ir jų prieigos atributai. Taip pat iškviečiami reikalingi funkcijoms IS servisai.
- *GrupesTeisesModelData* – klasė, kuri gražina užpildytą ar tuščią *GrupesTeisesModel* objektą.

- *GrupesTeisesModel* – klasė, kurioje saugoma visa informacija apie vartotojų teisių grupę.
- *GrupesTeisesListModel* – klasė, kurioje saugomas teisių grupių sąrašas.
- *GrupesTeisesListModelData* - klasė, kuri grąžina teisių grupių sąrašo objektą.
- *GrupesTeisesAddVartotojasCommand* – klasė, kuri paveldi bendrą IS komandos klasę ir atlieka vartotojų pridėjimo prie teisių grupės funkciją.
- *GrupesTeisesSaveCommand* - klasė, kuri paveldi bendrą IS komandos klasę ir atlieka teisių grupės duomenų saugojimo funkciją.
- *GrupesTeisesRemoveVartotojasCommand* - klasė, kuri paveldi bendrą IS komandos klasę ir atlieka vartotojų pašalinimo iš teisių grupės funkciją.

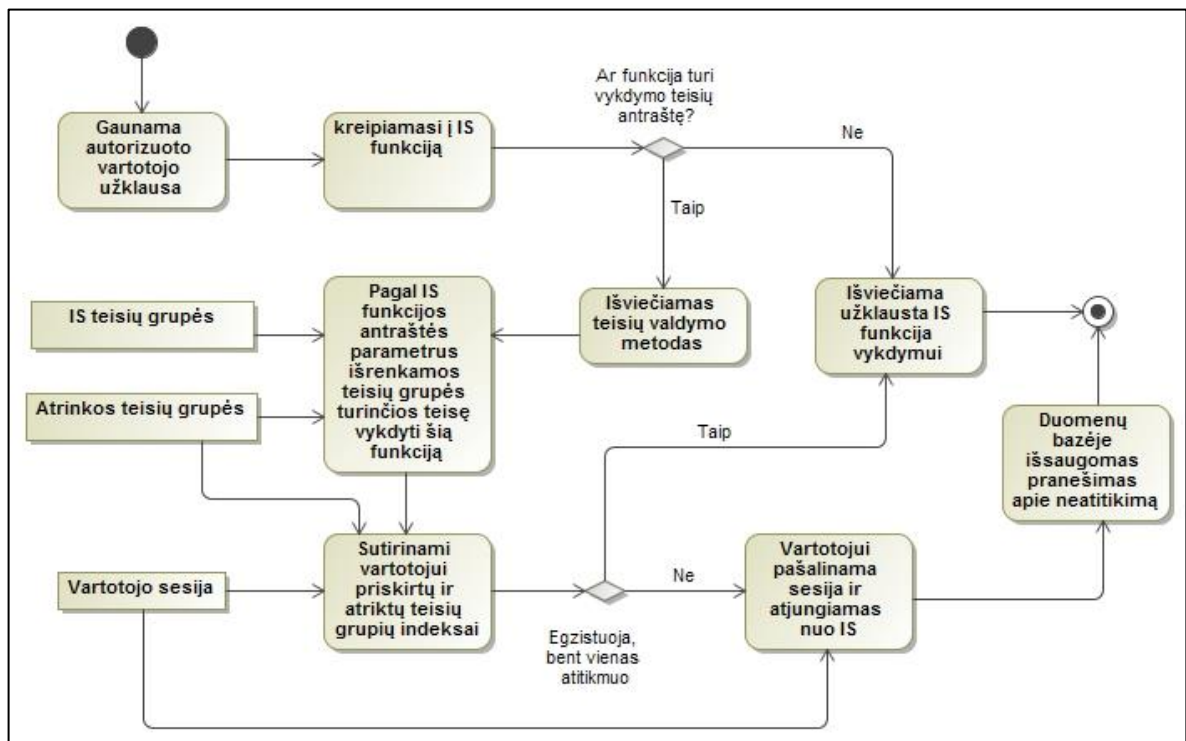
Sąsajai su duomenų baze panaudotas *Repository* šablonas, kuriame panaudotas *Entity Framework* įrankis. Toks sprendimas leido ženkliai optimizuoti atliekamų duomenų bazės užklausų kiekį. [8] LSMUSIS IS duomenų bazės fragmentas susijęs, su teisių valdymo metodu, pateiktas 13 paveiksle.



13 pav. LSMUSIS IS duomenų bazės fragmentas, susijęs su teisių valdymo metodu.

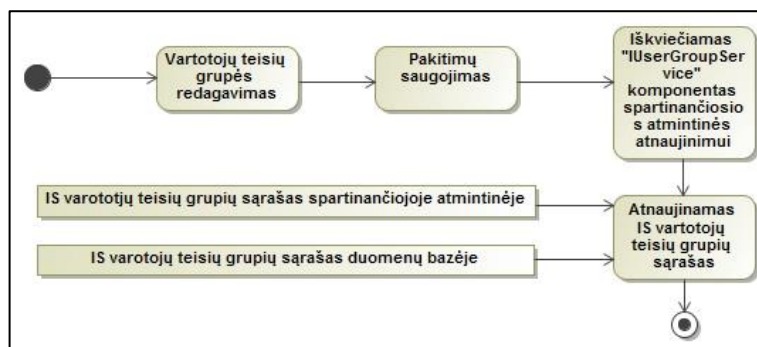
#### 4.5.2 Sistemos dinaminis vaizdas

LSMUSIS IS veiksmų sekos diagrama, atliekant vartotojo prieigos prie funkcijos teisių patikrinimą, pavaizduota 14 paveiksle. Kaip matome diagramoje, vartotojui prieš atliekant bet kokią IS funkciją, kuri reikalauja tam tikrų vartotojų teisių, yra išskviečiamas teisių valdymo metodas. Tai leidžia nutraukti užklausos vykdymą, net neiškvietus pačios funkcijos, jei vartotojas neturi reikiamų teisių jos vykdymui. Tokiu atveju, vartotojui pašalinama galiojanti sesija ir jis atjungiamas iš sistemos. Kiekviena tokia užklausa yra fiksuojama duomenų bazėje.



14 pav. Vartotojo prieigos prie IS funkcijos veiklos diagrama.

IS teisių valdymo administravimo sekų diagrama pateikta 15 paveiksle. Jame pagrindinis aspektas yra, kad išsaugojant atliktus pakeitimus vartotoju, teisių grupėje ( pridėjus ar pašalinus prieigas prie IS funkcijų), reikia iškviešti „*UserGroupsService*“ komponento metodą „*Update*“, kad atsinaujintų podėlyje esantys duomenys, nes prieigos prie funkcijos patikrinimo metodas vadovaujasi tik podėlyje esančiais duomenimis.



15 pav. IS vartotojų teisių grupių redagavimo veiklos diagrama.

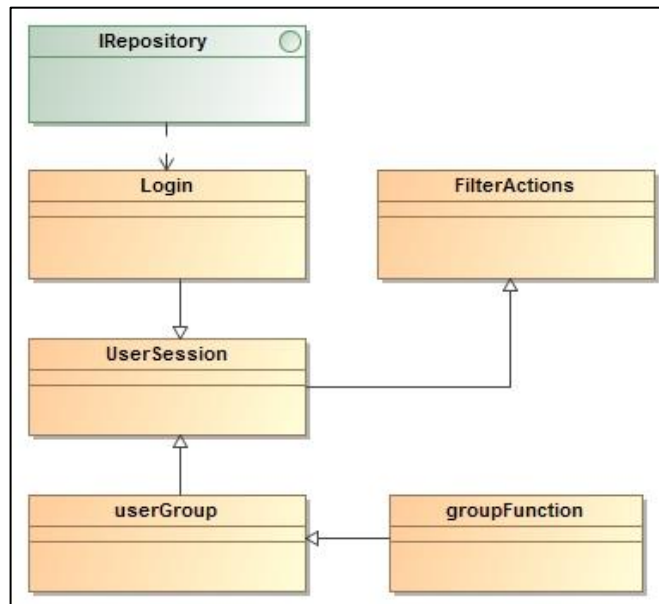
## 5 TYRIMO IR EKSPERIMENTINĖ DALIS

### 5.1 Atliekami tyrimai ir metodologija

Siekdami palyginti pasiūlyto vartotojų teisių valdymo metodo našumą su prieš tai realizuotu LSMUSIS IS atliksime tyrimus:

1. Sistemų atsako laiko priklausomybė nuo prisijungusių vartotojų skaičiaus.
2. Sistemoms reikiamos operatyvinės atminties kiekio priklausomybė nuo prisijungusių vartotojų skaičiaus.
3. Sistemų atsako laiko priklausomybė nuo turimų funkcijų skaičiaus.
4. Sistemoms reikiamos operatyvinės atminties kiekio priklausomybė nuo turimų funkcijų skaičiaus.
5. Sistemų atsako laiko priklausomybė nuo turimų vartotojų grupių skaičiaus.
6. Sistemoms reikiamos operatyvinės atminties kiekio priklausomybė nuo turimų vartotojų grupių skaičiaus.

Taip pat iš atliktų tyrimų mėginsime nustatyti sistemų atsako laiko ir užimamos operatyvinės atminties priklausomybę nuo trijų pagrindinių parametru: prisijungusių vartotojų kiekio, sistemos funkcijų ir teisių grupių skaičiaus.

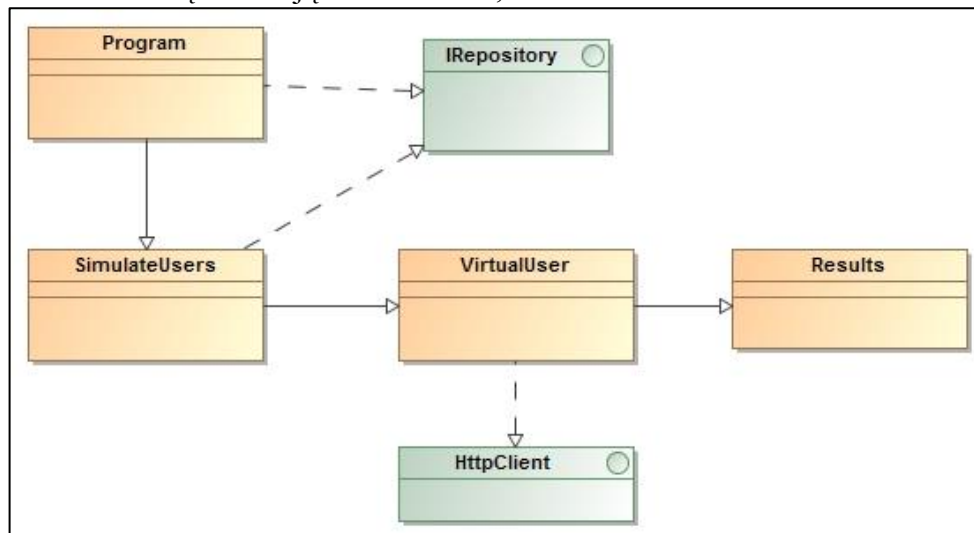


16 pav. Realizuoto egzistuojančio LSMUSIS IS teisių valdymo metodo klasių diagrama.

Siekiant gauti kuo tikslesnius matavimo rezultatus, metodai buvo realizuoti identiškoje aplinkoje, naudojant tą pačią duomenų bazę. Senojo LSMUSIS IS teisių valdymo metodo klasių diagrama pateikta 16 paveiksle, o siūlomo metodo struktūra aptarta projektinėje dalyje.

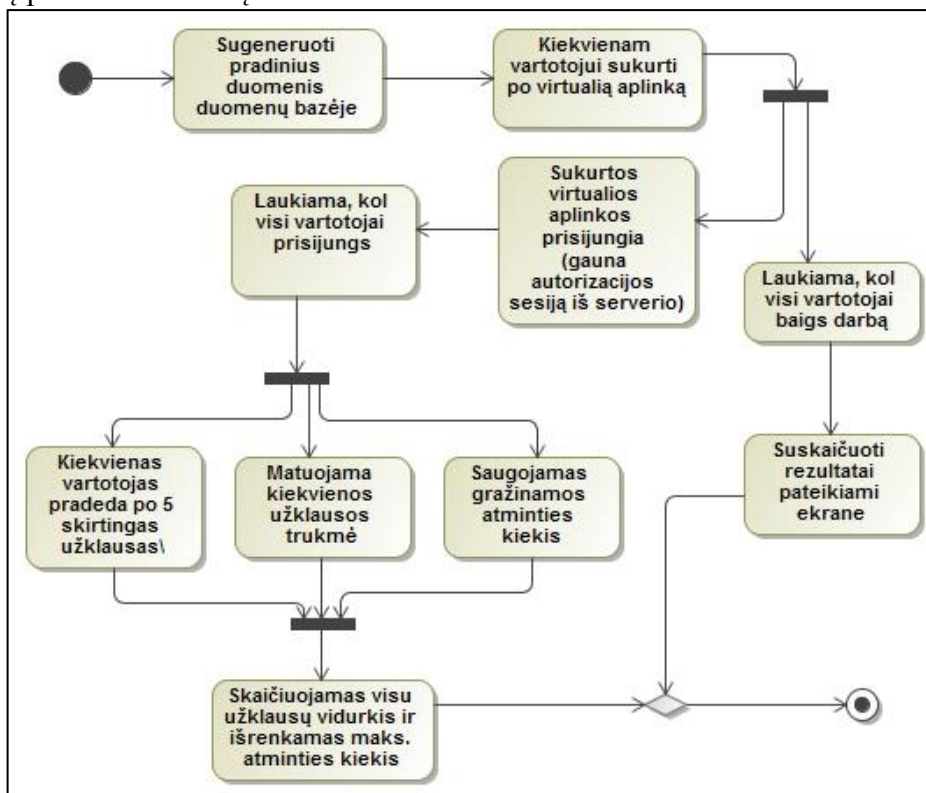
## 5.2 Tiriamosios programinės įrangos realizacija

Norint gauti matavimo rezultatus, kurie maksimaliai atitiktų realų sistemos veikimą, buvo realizuotas virtualių vartotojų simulatorius, kuris kartu atliko ir matavimo funkcijas.



17 pav. Realizuoto stimulatoriaus klasių diagrama.

Simulatorius, kiekvienam vartotojui sukuria individualią giją, kuri atitinka virtualų vartotoją, kuris jungiasi prie sistemos su unikaliu prisijungimo vardu ir slaptažodžiu. Prisijungimo metu jis yra autorizuojamas, jam suteikiamos privilegijos pagal vartotojui priskirtas teisių grupes. Po sėkmingo prisijungimo virtualus vartotojas identifikuoja save kaip prisijungusį ir pasirengusį pradėti matavimą.



18 pav. Simulatoriaus matavimų procesas.



Kai visos gijos prisijungia ir pasiruošia matavimui, centrinis procesas visoms gijoms duoda komandą pradėti. Visi vartotojai, gavę komandą inicijuoja 5 skirtingas užklausas, kurios atitinka 5 skirtingas funkcijas ir kurios turi skirtingas prieigos teises. Tad kreipiantis į jas, kiekvienai jų reikia patikrinti, ar vartotojas turi reikiamas privilegijas jos vykdymui. Pačių funkcijų turinys yra identiškas, jų gautas rezultatas yra užimamos atminties kiekis baitais. Kiekviena gija fiksuoją užtruką užklauso laika vienos užklauso metu ir gaunama užimtos atminties kiekį.

Baigus matavimą yra išrenkamas maksimalus užimamos atminties kiekis ir suskaičiuojama vidutinė užklauso trukmė. Visi gauti rezultatai yra tiesiogiai priklausomi nuo naudojamos programinės ir aparatūrinės įrangos, tad matavimai atlikti prie kitokios sistemos konfigūracijos gali neatitikti ar net iškreipti pateiktas išvadas.

### 5.3 Tyrimui naudojamos įrangos bei parametrų aprašas

Visi tyrimai atlikti naudojantis pateikta programine ir aparatūrine įranga:

- Procesorius: Intel Xeon E3 1245 v3 @3.6 GHz
- Centrinis lustas: Intel H87
- Kietasis diskas: Corsair Force GT 120GB SSD
- Operatyvioji atmintis: Kingston HyperX DDR3 8 GB @1600 MHz
- Operacinė sistema: Microsoft Windows 7 64 bitų
- SQL serveris: Microsoft SQL Server 2008
- Http serveris: Microsoft IIS 7.5
- Karkasas: Microsoft .NET Framework 4.5.1

Sistemos parametrai, kurie buvo naudojami tyrime suskirstyti į 3 grupes:

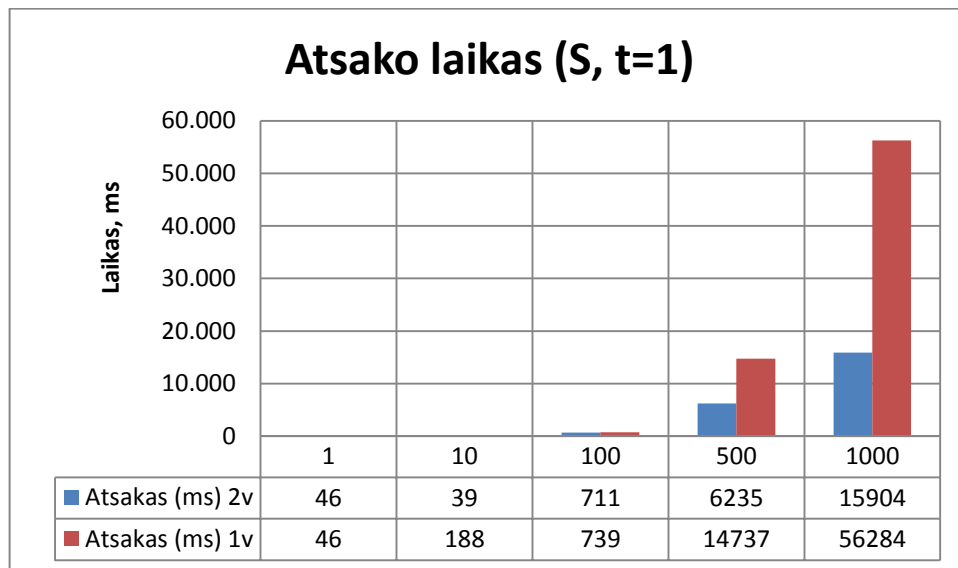
#### 1 lentelė. IS dydžių parametrai.

Sistemos dydis	S (mažas)	M (vidutinis)	L (didelis)
Prisijungusių vartotojų kiekis	10	100	1000
Sistemos funkcijų kiekis	10	100	500
Vartotojų grupių kiekis sistemoje	5	50	100

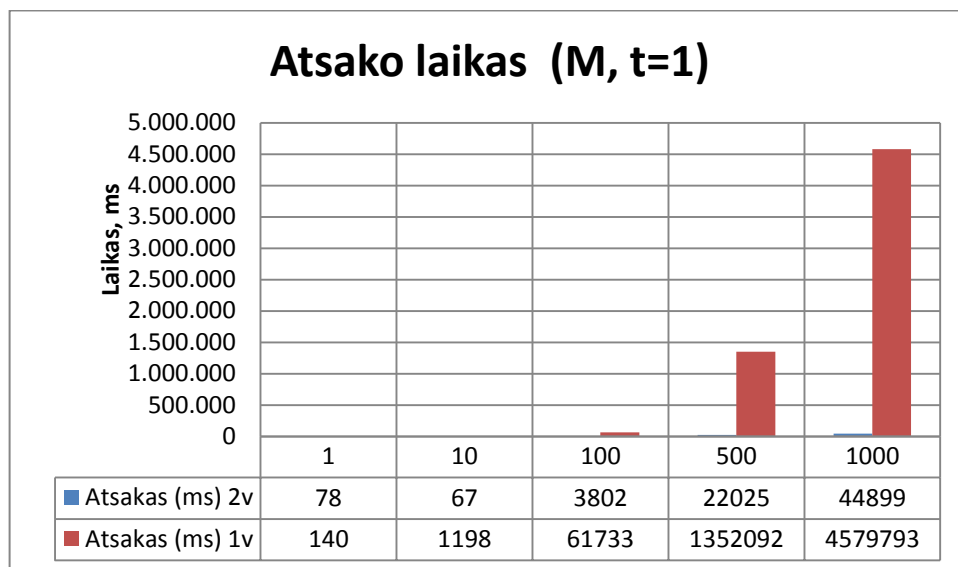
### 5.4 Eksperimentinių tyrimų rezultatai

#### 5.4.1 Sistemų atsako laiko priklausomybės nuo prisijungusių vartotojų skaičiaus tyrimas.

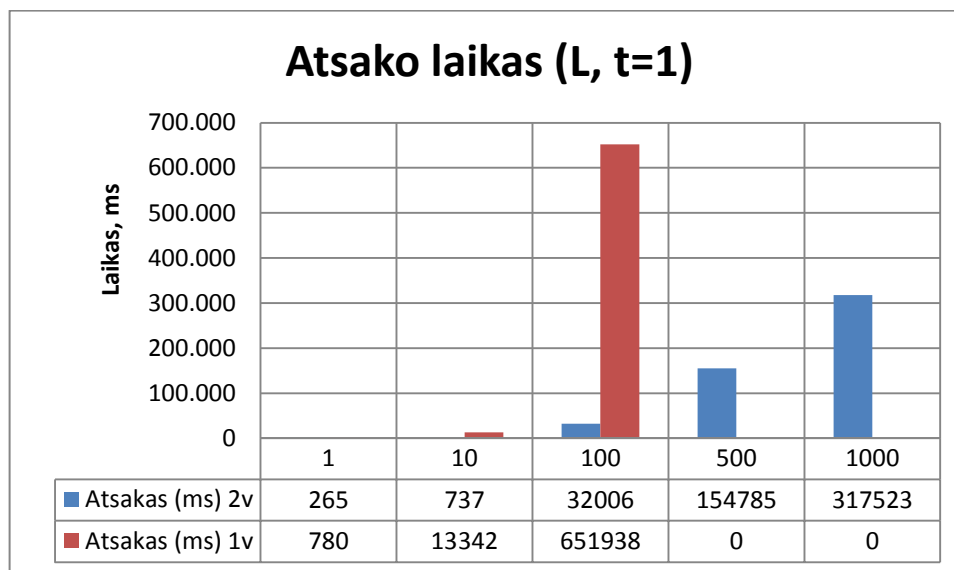
Šiame tyrime nustatysime, kaip prisijungusių vartotojų kiekis lemia sistemos atsako laiką, prie skirtingų sistemos dydžių. Atlikome po 5 matavimus kiekvienai iš jų. Gauti rezultatai pateikti 19-21 paveiksluose.



19 pav. Atsako laiko priklausomybė nuo vartotojų skaičiaus, esant S dydžio sistemai.



20 pav. Atsako laiko priklausomybė nuo vartotojų skaičiaus, esant M dydžio sistemai.

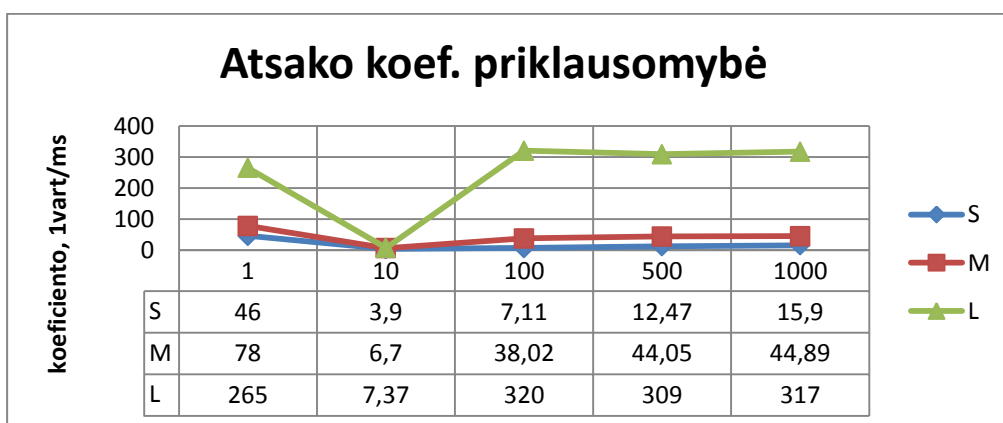


**21 pav. Atsako laiko priklausomybė nuo vartotojų skaičiaus, esant L dydžio sistemai.**

Atlikę matavimus matome, kad sistemos atsako laikas yra priklausomas nuo prisijungusių vartotojų aktyvumo. Iš grafiko galime teigti, kad sistemos su pasiūlytu metodu atsako laikas nesiskiria prie mažos apkrovos, bet yra net kelis kartus mažesnis prie didelės. S ir M dydžio sistemose matome netiesioginę priklausomybę nuo apkrovos. Tai lemia skaičiavimo metodiką. Esant vienam vartotojui sistema užtrunka, kol užsikrauna duomenis į operatyvią atmintį ir tai padidina atsako laiką. Kai yra daugiau negu viena užklausa, sueikvotas laikas užrovimui tampa aktualus tik pirmosioms užklausoms, visos kitos jau naudoja užkrautus duomenis, tad jų vykdymo trukmė yra ženkliai mažesnė. Bendras atsako trukmės vidurkis prie didesnės apkrovos yra mažesnis, nei prie minimalios.

Esant didelės apimties (L) sistemai, prie didelio vartotojų antplūdžio, sistema pritrūksta operatyvios atminties ir pradeda rašyti duomenis į kietą diską, tai lemia labai ženklių spartos kritimą, tad nuspręsta rezultatus prie tokių sąlygų laikyti netinkamais.

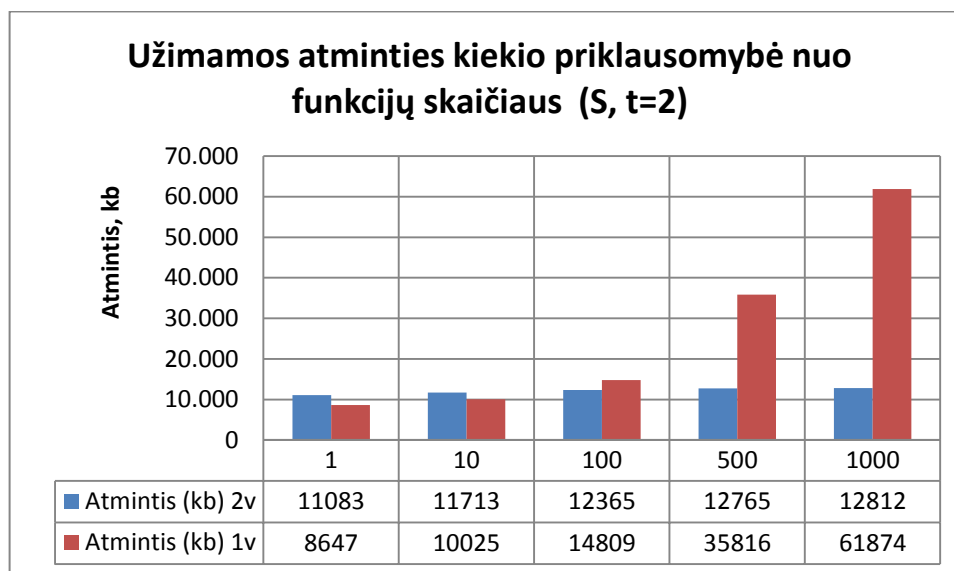
Iš šio tyrimo galime teigti, kad pasiūlytas teisių valdymo metodas pasitvirtina tiek mažose, tiek didelėse sistemose, esant skirtingam kiekiui vartotojų. Išreiškus atsako laiko priklausomybę nuo vartotojų skaičiaus (koeficientas = atsako laikas / vartotojų skaičius) 7 paveiksle galime išvelgti tendencijas ir sudaryti formulę  $t$  (atsako laikas) =  $n$  (vartotojų skaičius) \*  $k$  (kintamasis, kurį bandysime nustatyti 5.3.3 ir 5.3.5 skyriuose).



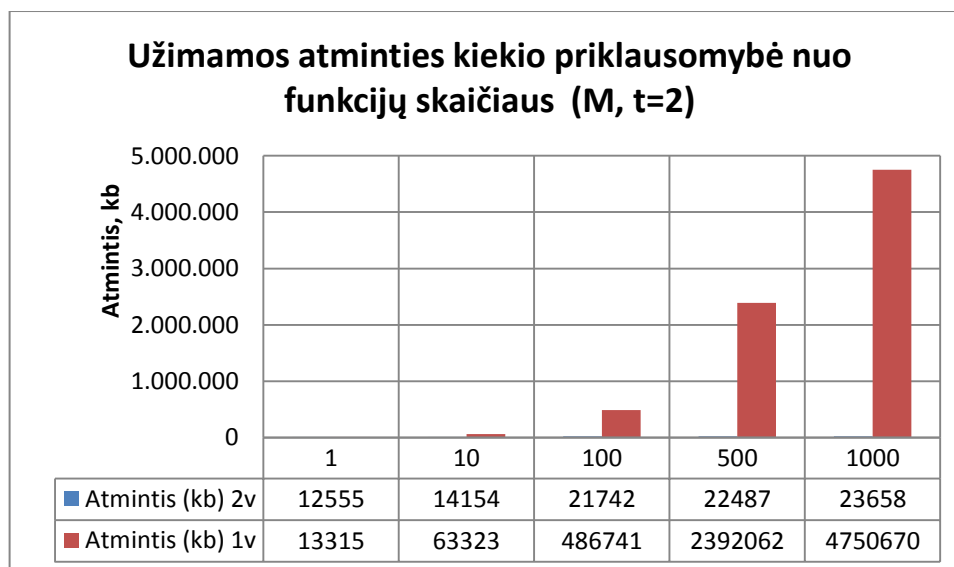
**22 pav. Atsako laiko koeficiento priklausomybė nuo vartotojų skaičiaus.**

#### 5.4.2 Sistemoms reikiamos operatyvinės atminties kiekio priklausomybės nuo prisijungusių vartotojų skaičiaus tyrimas.

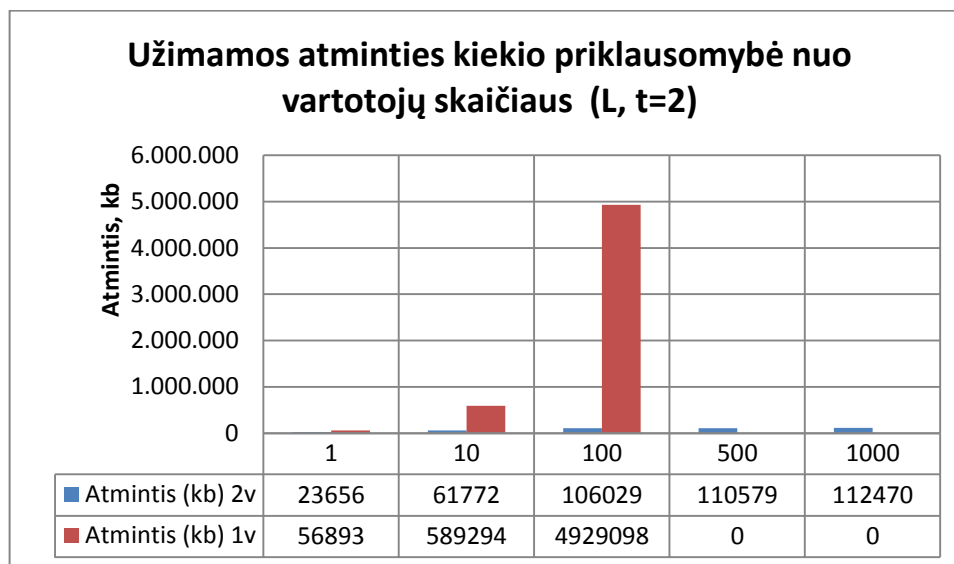
Šiame tyrime nustatysime, kaip prisijungusių vartotojų kiekis lemia sistemos užimamą atminties kiekį prie skirtingų sistemos dydžių. Atlikome po 5 matavimus kiekvienai iš jų. Gauti rezultatai pateikti 23-25 paveiksluose.



23 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų skaičiaus, esant S dydžio sistemai.



24 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų skaičiaus, esant M dydžio sistemai.

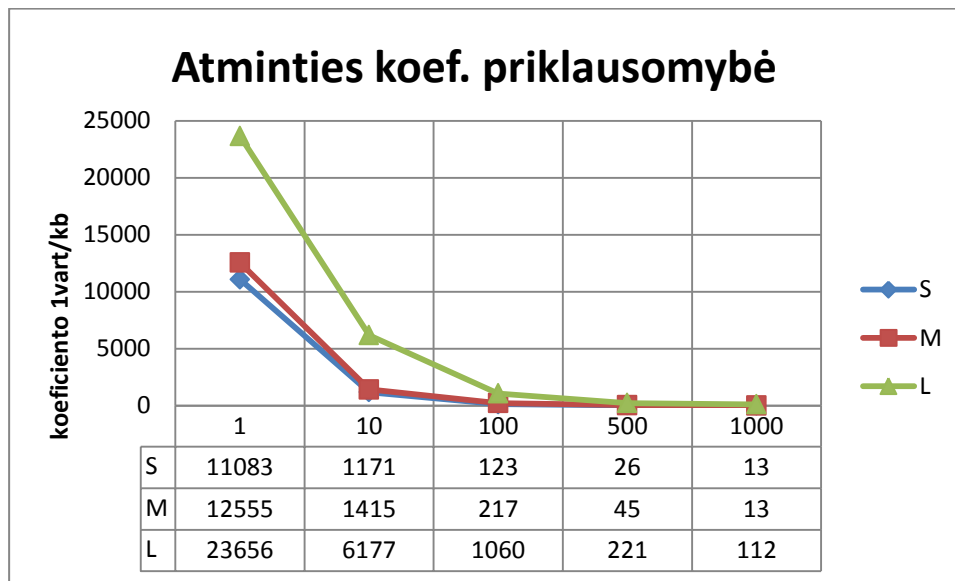


**25 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų skaičiaus, esant L dydžio sistemai.**

Atlikę matavimus tarp skirtingų sistemų, galime išvelti vieną esminį skirtumą. Su pasiūlytu teisių prieigos metodu užimama atmintis išlieka panaši ir didėja neženkliai nuo prisijungusių vartotojų skaičiaus. Tuo tarpu sistemai su senuoju LSMUSIS IS metodu reikalingas atminties kiekis yra proporcingas vartotojų skaičiui ir didėja gan ženkliai, didėjant naudotojų skaičiui. Tokį didėjimą sąlygoja tai, kad kiekvienam vartotojui prisijungimo metu yra suteikiama sesija, kurioje yra talpinama visos vartotojų leidžiamų funkcijų prieigos, tad esant dideliame sistemos funkcionalumui reikalingas itin didelis kiekis atminties.

Sistema su pasiūlytu metodu reikalauja gerokai mažesnio kiekio atminties priklausomai nuo funkcionalumo, bet išlieka proporcinga jų kiekiui. Tokius rezultatus lemia maksimaliai sumažintas informacijos kiekis, saugojamas vartotojo sesijoje, o visos prieigos saugomos servise nedubliuojant įrašų per skirtingus vartotojus ir jų grupes. Tad priklausomybė nuo vartotojų skaičiaus kinta labai nežymiai, nes kiekvienas vartotojas, prisijungęs serveryje, pareikalauja papildomos atminties tik identifikavimui ir jam priklausančių vartotojų grupių indeksams.

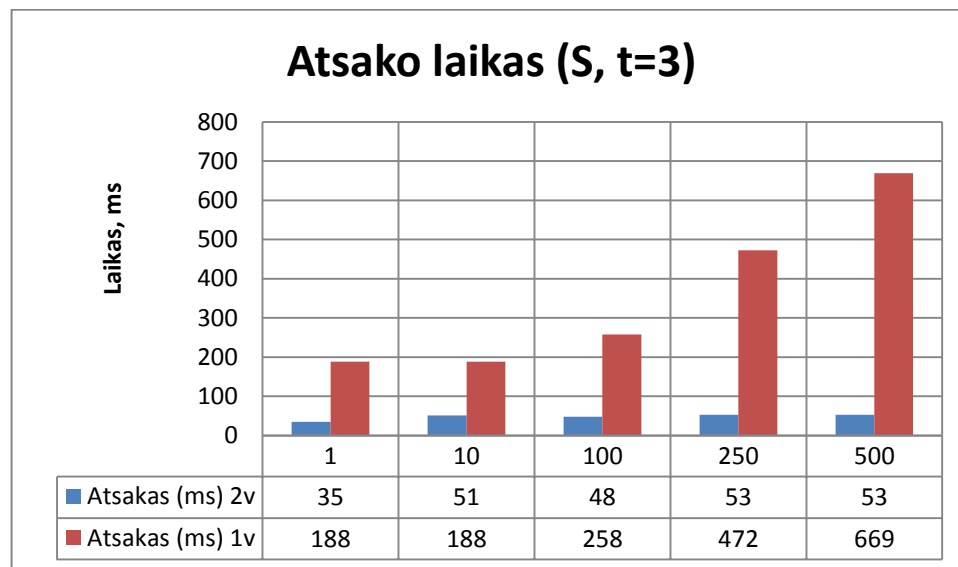
Siekdami nustatyti reikiamą sistemos atminties kiekį, numatydami būsimas apkrovas, apskaičiuojame išskirtos atminties kiekį vienam vartotojui, koeficientus prie skirtingų dydžių sistemų. Iš pateikto grafiko (26 pav.) galime teigti, kad užimamos atminties kiekis yra priklausomas nuo prisijungusių vartotojų skaičiaus ir sistemos dydžio.



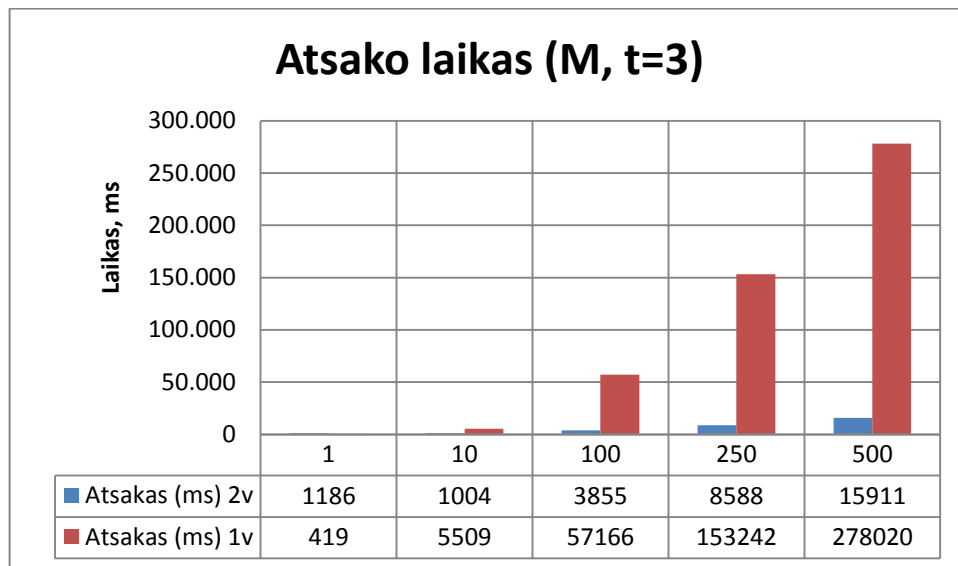
26 pav. Užimamos atminties koeficiento priklausomybė nuo vartotojų skaičiaus.

#### 5.4.3 Sistemų atsako laiko priklausomybė nuo turimų funkcijų skaičiaus tyrimas.

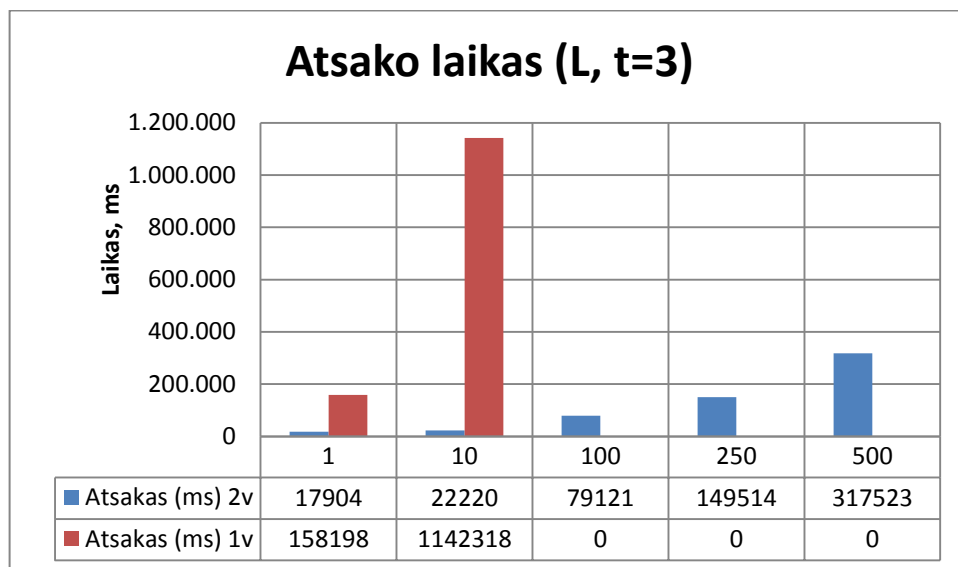
Šiame tyrime nustatysime, kaip funkcijų kiekis lemia atsako laiką prie skirtingų sistemos dydžių. Atlikome po 5 matavimus kiekvienai iš jų. Gauti rezultatai pateikti 24-29 paveiksluose.



27 pav. Atsako laiko priklausomybė nuo funkcijų skaičiaus, esant S dydžio sistemai



28 pav. Atsako laiko priklausomybė nuo funkcijų skaičiaus, esant M dydžio sistemai



29 pav. Atsako laiko priklausomybė nuo funkcijų skaičiaus, esant L dydžio sistemai

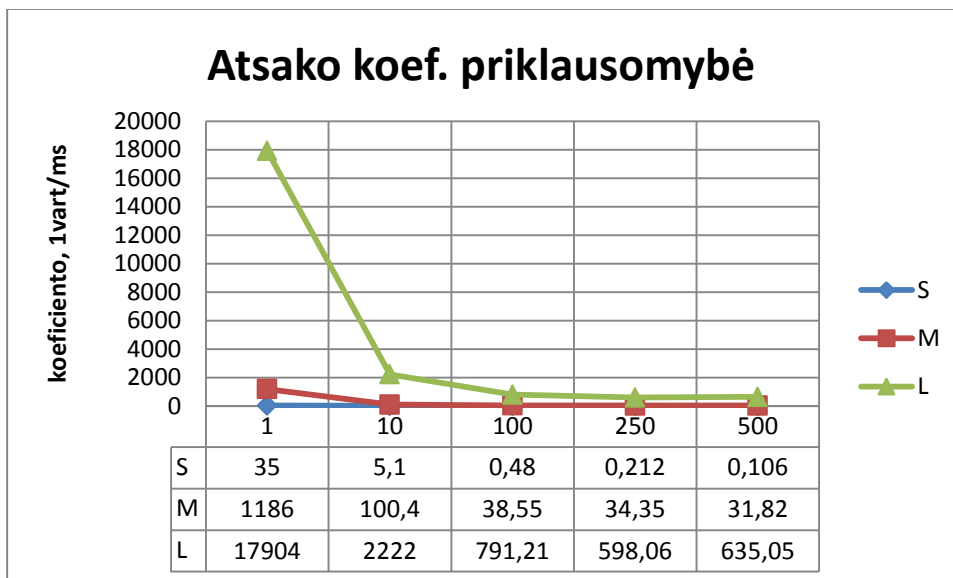
Atlikę matavimus iš gautų rezultatų galime teigti, kad atsako laikas yra priklausomas nuo sistemos funkcijų skaičiaus, ypač didesnės apimties sistemose. Lygindami siūlomą su senuoju metodu, matome didelius skirtumus prie didesnio funkcionalumo (100 ir daugiau). Tai įrodo siūlomo sprendimo našumo pranašumą prie itin didelio funkcijų kiekio.

Vienas iš pagrindinių architektūrinių sprendimų, kuris padėjo gauti tokius rezultatus, tai funkcijų priklausomybių grupėms nuolatinis saugojimas atmintyje ir maksimali duomenų indeksacija. Tokiu atveju, tikrinant prieigos galimybę prie funkcijos, yra lyginami tik du indeksų masyvai:

- vartotojų sesijoje saugojamos jam priskirtų teisių grupių indeksų masyvas
- grupių indeksų masyvas, kurios turi prieigą prie tikrinamos funkcijos

Kadangi tyrimo metu pastebėjome, kad atsako laikas yra proporcingas funkcijų kiekiui sistemoje, galime patobulinti pirmame tyrime pateiktą atsako laiko skaičiavimo formulę. Tam

suskaičiavome vienos funkcijos priklausomybės atsako laikui koeficientą skirtingų dydžių sistemose. Rezultatai pateikti 30 pav.



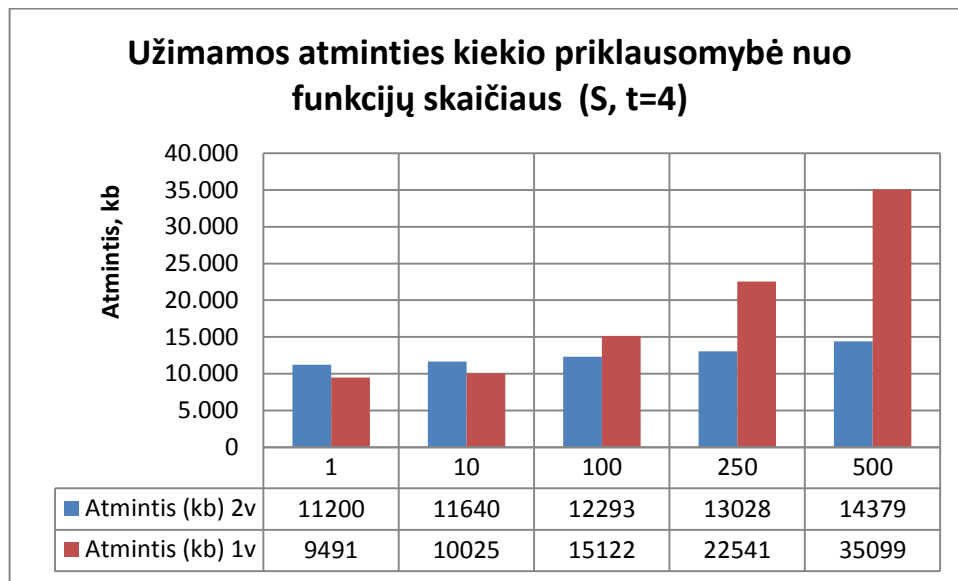
**30 pav. Atsako laiko koeficiento priklausomybė nuo funkcijų skaičiaus sistemoje.**

Iš pateikto grafiko matome, kad koeficientas tolygiai mažėja, didėjant sistemos funkcijų skaičiui (esant dideliui sistemos funkcionalumui). Turimoje formulėje pridedam funkcijų skaičiaus dedamąją  $t = n * k * (0,0175 * f)$ . Daugiklį prie  $f$  susiskaičiuojame pagal tiesės lygtį, kurią susidarome remdamiesi 30pav. Tiesė brėžiama tarp taškų: 100, 250 ir 500.

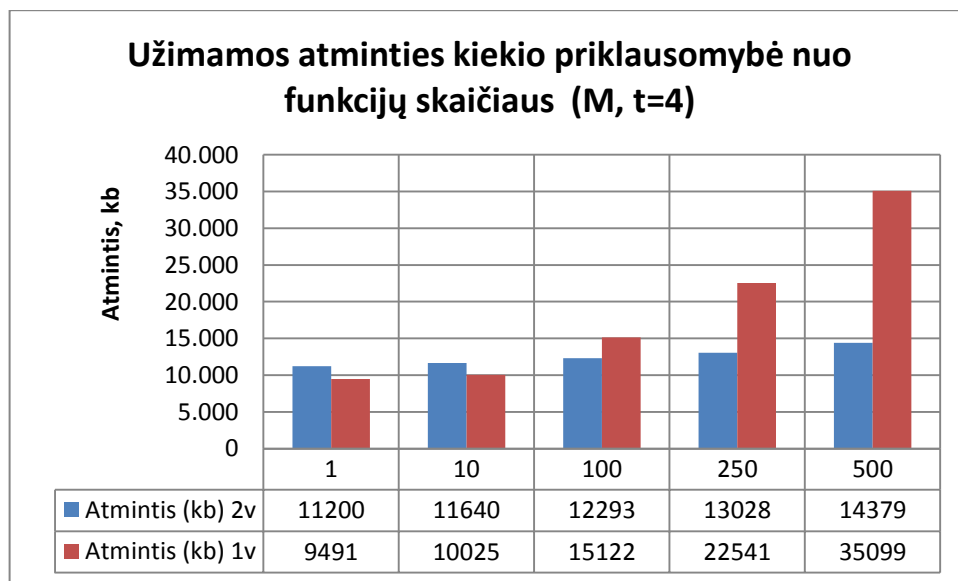
#### 5.4.4 Sistemoms reikiamos operatyvinės atminties kiekio priklausomybės nuo turimų funkcijų skaičiaus tyrimas.

Šiame tyrime nustatysime, kaip sistemos funkcijos lemia, sistemos užimamos atminties kiekį prie skirtingų sistemos dydžių. Atlikome po 5 matavimus kiekvienai iš jų. Gauti rezultatai pateikti 31-33 paveiksluose.

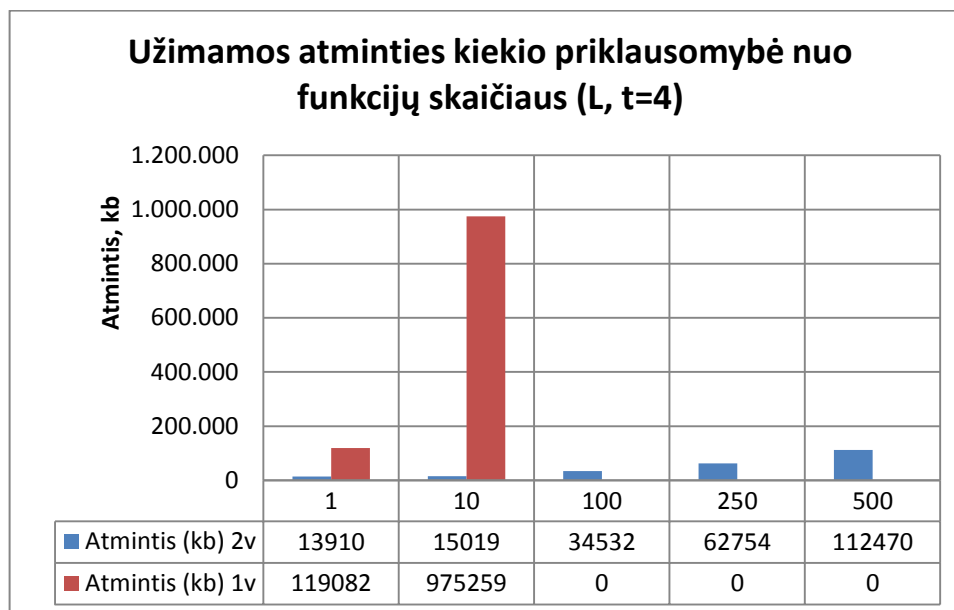




31 pav. Sistemos užimamos atminties priklausomybė nuo funkcijų skaičiaus, esant S dydžio sistemai.

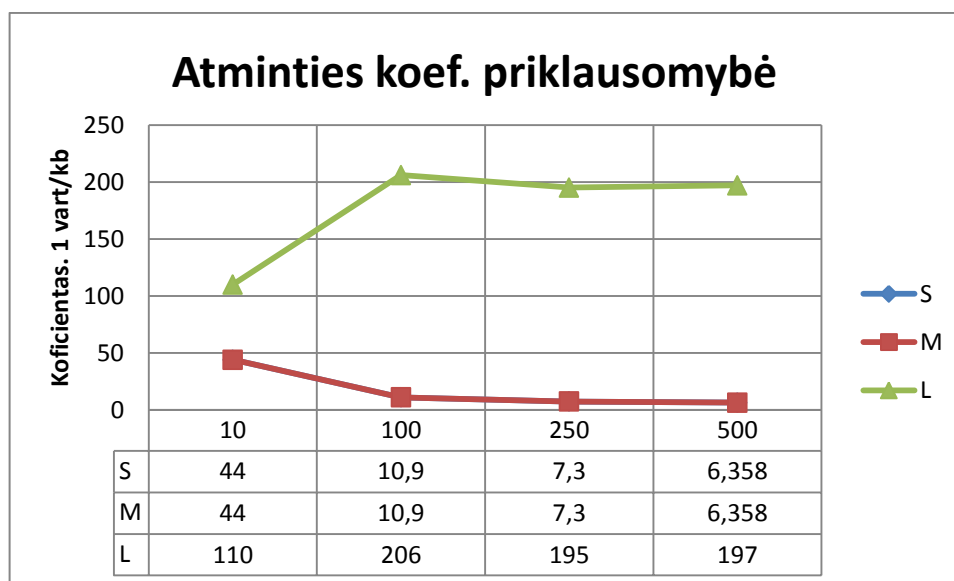


32 pav. Sistemos užimamos atminties priklausomybė nuo funkcijų skaičiaus, esant M dydžio sistemai.



**33 pav. Sistemos užimamos atminties priklausomybė nuo funkcijų skaičiaus, esant L dydžio sistemai.**

Iš gautų rezultatų pastebime, kad abiem sistemoms reikalingas operatyvios atminties kiekis priklauso nuo funkcijų gausos. Be to pastebimas ir ženklus priklausomybės skirtumas, tad vėl galime teigti, kad pasiūlytas teisių valdymo metodas yra daug efektyvesnis ir reikalauja mažiau atminties funkcijai. Remdamiesi šiais grafikais, sudarome vienos funkcijos užimamos atminties kiekio priekingų dydžių sistemų priklausomybės grafiką (34 pav.).

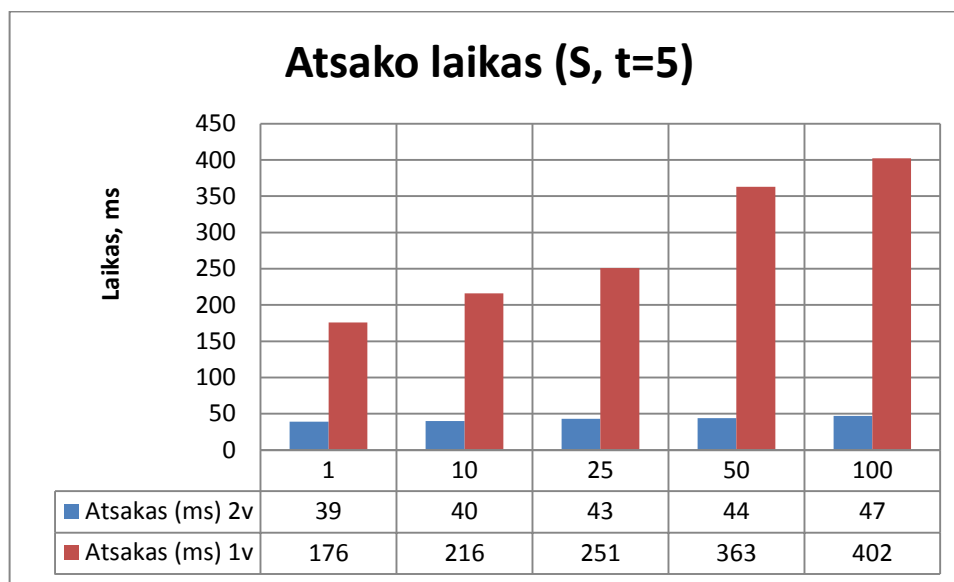


**34 pav. Užimamos atminties koeficiento priklausomybė, nuo funkcijų skaičiaus.**

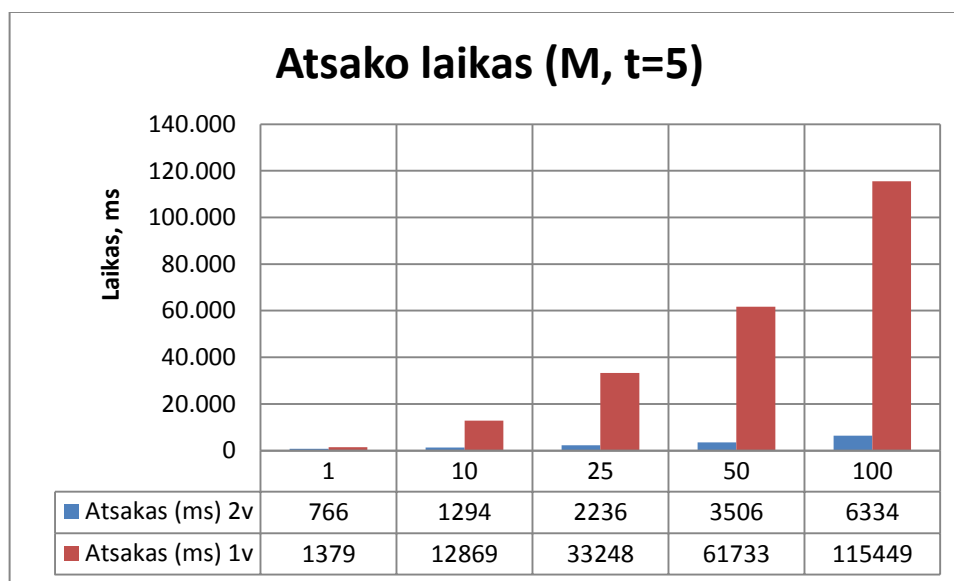
Remdamiesi šiuo grafiku, galime teigti, kad, didėjant funkcionalumui, vienai funkcijai reikalingas atminties dydis, nors ir nežymiai, bet mažėja.

#### 5.4.5 Sistemų atsako laiko priklausomybė nuo turimų vartotojų grupių skaičiaus tyrimas.

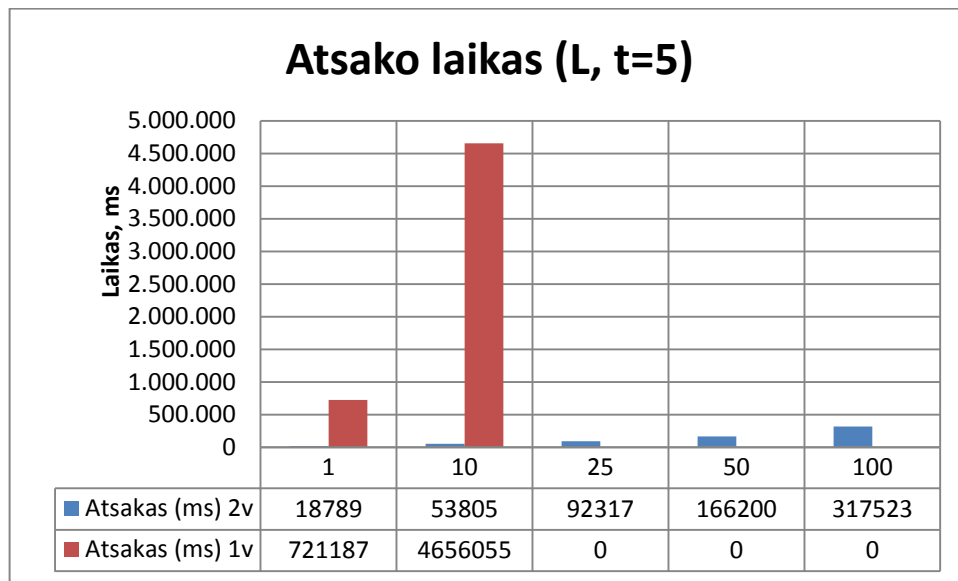
Šiame tyrime nustatysime, kaip vartotojų grupių kiekis lemia atsako laiką prie skirtingų sistemos dydžių. Atlikome po 5 matavimus kiekvienai iš jų. Gauti rezultatai pateikti 35-37 paveiksluose.



35 pav. Atsako laiko priklausomybė nuo vartotojų grupių skaičiaus, esant S dydžio sistemai.



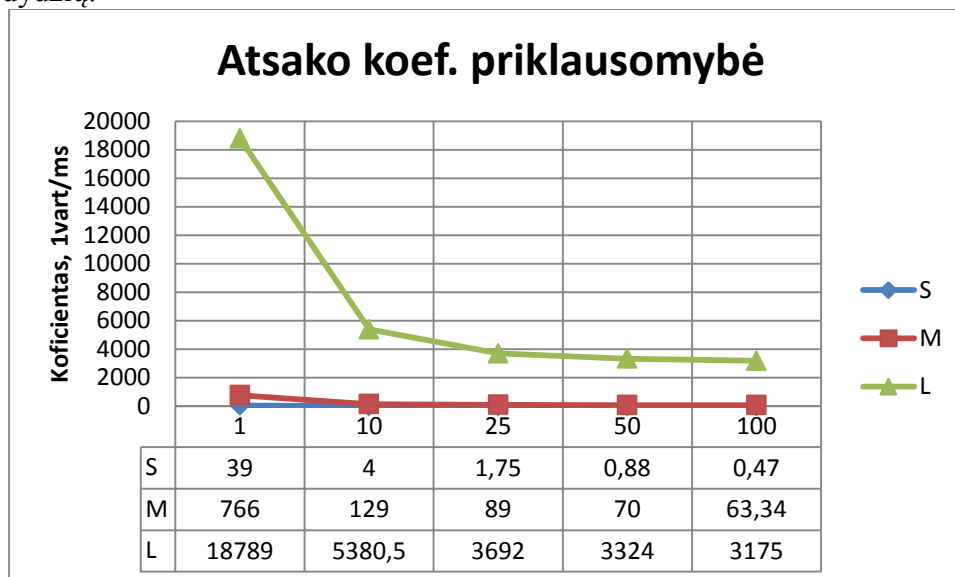
36 pav. Atsako laiko priklausomybė nuo vartotojų grupių skaičiaus, esant M dydžio sistemai.



**37 pav. Atsako laiko priklausomybė nuo vartotojų grupių skaičiaus, esant L dydžio sistemai.**

Iš gautų tyrimo rezultatų galime teigti, kad vartotojų grupių skaičius lemia sistemos atsako laiko abiejose sistemose, bet atsako laiko priklausomybė nuo grupių yra nevienoda. Vėlgi galime daryti išvadą, kad pasiūlytas teisių valdymo sprendimo metodas yra našesnis, nes įtakos dydis yra mažesnis. Tam didžiausią įtaką daro, kad, sutikrinant prieigos galimybę prie funkcijos, vartotojui yra sutikrinami tik indeksų masyvai, kurie yra saugojami vartotojo sesijoje ir teisių servise, kur visi duomenys yra saugojami operatyvinėje atmintyje.

Atlikus visus tris tyrimus, susijusius su atsako laiku, galime teigti, kad atsako laikas pasiūlytame metode yra priklausomas nuo vartotojų, funkcijų sistemoje ir grupių skaičiaus. Taip pat galime užbaigti 1 tyrime pradėtą ir 3 papildytą formulę, prie kurios dar pridėsime grupių skaičiaus sistemoje dedamąją ir patikrinsim jos teisingumą. Pirmiausiai nusistatysime dedamąją iš gautų tyrimo rezultatų, sudarę kiekvienos papildomos grupės įtaką atsako laikui prie įvairių sistemų dydžių.



**38 pav. Atsako laiko koeficiento priklausomybė nuo vartotojų grupių skaičiaus sistemoje.**

Grafike matome grupių skaičiaus priklausomybės kitimą, keičiantis vartotojų grupių skaičiui. Tad prie turimos formulės pridėdami grupių dedamąją su daugikliu, nustatytu iš grafiko. Gausime numatomo atsako laiko skaičiavimo metodiką  $t = n * k * (0,0175 * f) * (0,14 * g)$ . Supaprastinę gauname „ $t = n * k * f * g$ “, kur k prie matuotos programinės įrangos yra 0,00637. Prie kiekvienos sistemos reikia nusistatyti k naudojant simulatorių. Gautai išraiškai patikrinti sudarome lentelę:

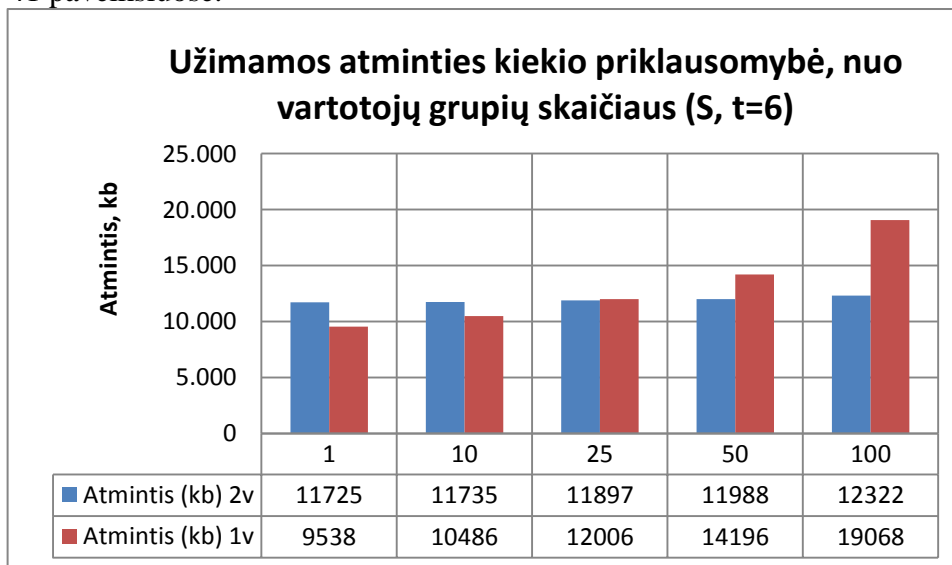
**2 lentelė. Gautų ir numatomų rezultatų skirtumai.**

n	f	g	n*f*g*k	Išmatuotas	Skirtumas	Paklaida %
100	500	100	31850	32006	156	0,487408611
1000	500	100	318500	317523	-977	-0,307694246
500	500	100	159250	154785	-4465	-2,884646445
1000	250	100	159250	149514	-9736	-6,511764785
1000	100	100	63700	79121	15421	19,49040078
100	500	50	15925	15911	-14	-0,087989441
1000	500	25	79625	92317	12692	13,74828038
1000	500	50	159250	166200	6950	4,181708785
100	100	25	1592,5	2236	643,5	28,77906977

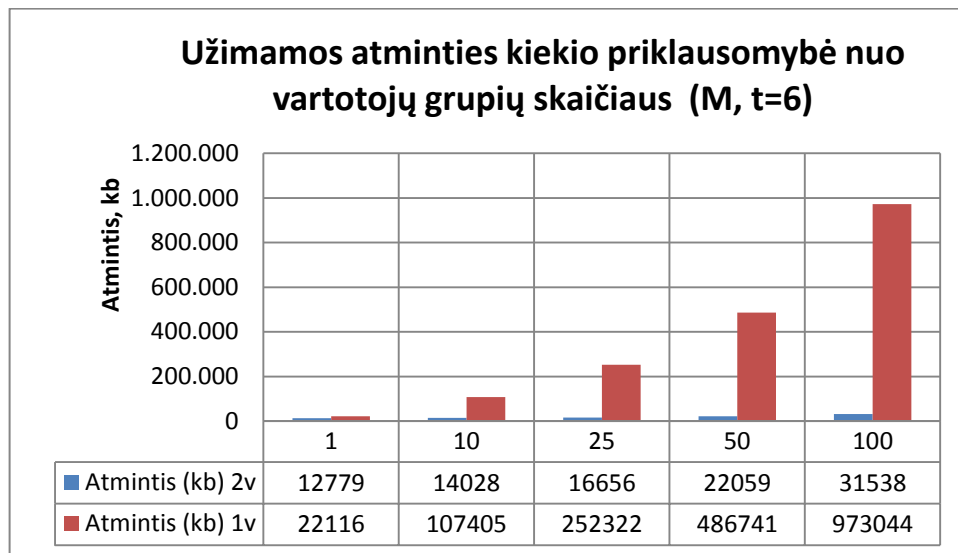
Iš jos matome, kad tiesinė tiesioginė priklausomybė galioja tik esant didelėms apkrovoms ir gausiam funkcijų ir vartotojų grupių skaičiui. Prie mažų parametrų paklaida siekia net 28%.

#### 5.4.6 Sistemoms reikiamos operatyvinės atminties kiekio priklausomybė nuo turimų vartotojų grupių skaičiaus tyrimas.

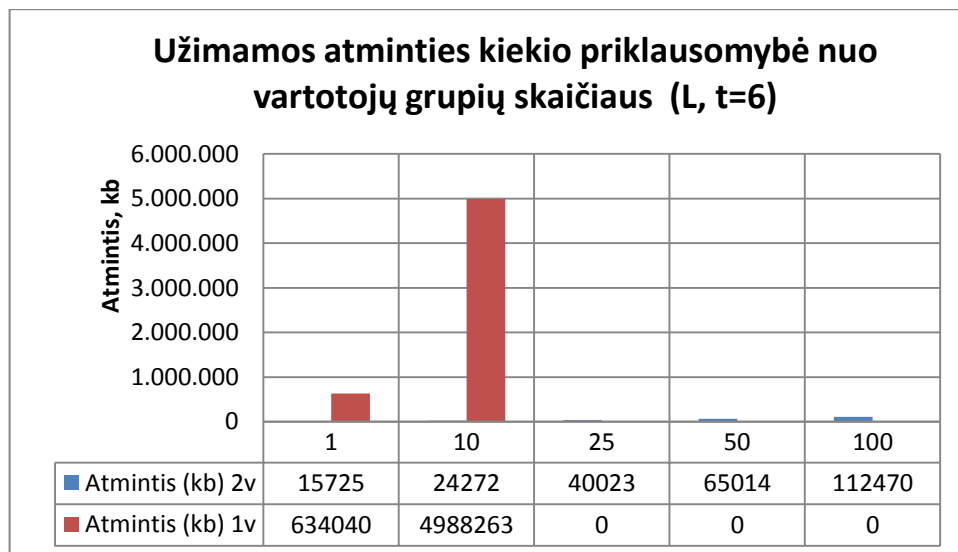
Šiame tyrime nustatysime, kaip sistemos funkcijos lemia sistemos užimamos atminties kiekį prie skirtingų sistemų dydžių. Atlikome po 5 matavimus kiekvienai iš jų. Gauti rezultatai pateikti 39-41 paveiksluose.



**39 pav. Sistemos užimamos atminties priklausomybė, nuo vartotojų grupių skaičiaus esant S dydžio sistemai.**

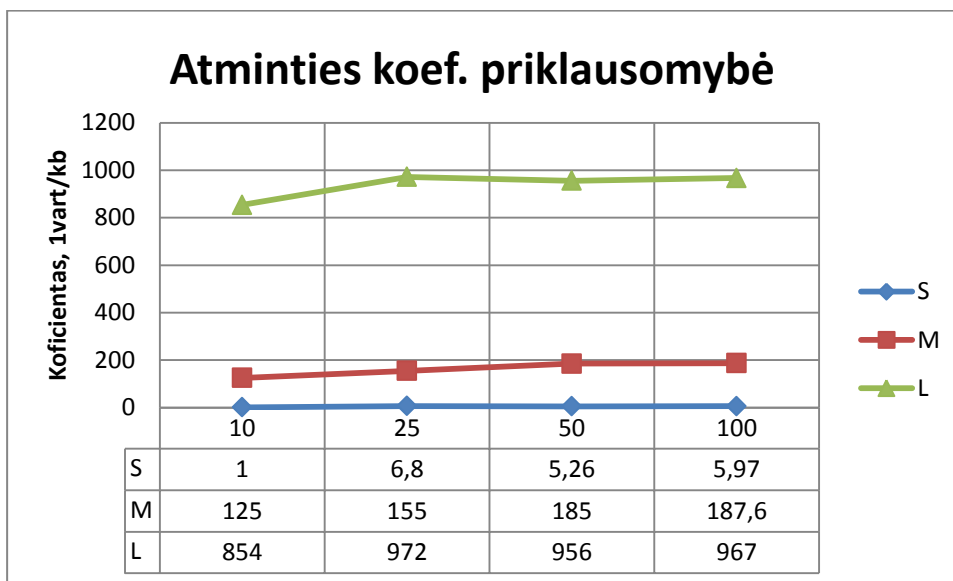


**40 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų grupių skaičiaus, esant M dydžio sistemai.**



**41 pav. Sistemos užimamos atminties priklausomybė nuo vartotojų grupių skaičiaus, esant L dydžio sistemai.**

Iš gautų rezultatų matome, kad vartotojų grupių kiekis lemia sistemos užimamos operatyvinės atminties kiekį abejuose sistemose, kaip ir kiti, jau išmatuoti parametrai. Tai parodo, kad prisijungus vartotojui, jam priskiriamos grupės yra saugomos sesijoje. Kuo daugiau grupių turi vartotojas, tuo jo sesija reikalauja daugiau vietos. Visi matuoti dydžiai yra tarpusavyje susiję ir proporcingai nulemia užimamos atminties kiekį.



**42 pav. Užimamos atminties koeficiento priklausomybė nuo vartotojų grupių skaičiaus.**

Remdamiesi pateiktu grafiku ir prieš tai atliktais tyrimais galime teigti, kad sistemos užimamos atminties kiekis yra labai priklausomas nuo dviejų susijusių parametų: vartotojų kiekio ir jiems priskirtų grupių skaičiaus. Tai yra todėl, kad kiekvieno vartotojo sesija atmintyje išskiria vietas visoms savo grupėms aprašyti, o jos ir funkcijos yra saugomos atmintyje jų nedubliuojant per visus vartotojus.

## 6 IŠVADOS

1. Atlikus egzistuojančių teisių valdymo metodų analizę, nuspręsta pasiūlyti tokį metodą, kuriame būtų panaudoti sujungti geriausi išanalizuotų metodų bruožai, o siūlomo metodo našumas ir funkcionalumas atitiktų visus LSMUSIS IS specifikacijos reikalavimus.
2. Išanalizavus skaičiavimų našumą gerinančius įrankius ir karkasus skirtus ASP.NET MVC karkasui, nuspręsta panaudoti podėlį dažniai atliekamų identiškų skaičiavimų rezultatams išsaugoti. Realizuota metodo architektūra išnaudoja podėlį dažnai naudojamiems duomenų rinkiniams saugoti.
3. Atlikus atsako laiko charakteristikų tyrimą (5.4.1, 5.4.3 ir 5.4.5 tyrimai) ir išanalizavus gautus rezultatus nustatyta, kad pasiūlytas vartotojų teisių valdymo metodas yra spartesnis ir panaudoja mažiau operatyvios atminties nei buvęs metodas LSMUSIS sistemoje, ypač esant vidutinėm (M) ir didelėm (L) sistemoms (M ir L dydžių aprašymas pateiktas 5.3 skyriuje 1 lentelėje). Tai parodo atliktų tyrimų rezultatai, kuriuose gautas sistemos atsako laikas mažesnis, net iki 86 kartų.
4. Atliktų reikiamos operatyvios atminties charakteristikų tyrimų (5.4.2, 5.4.4 ir 5.4.6 tyrimai) rezultatai rodo, kad pasiūlytas metodas yra našesnis už liktinės sistemos. Atlikus tyrimą (5.4.6 tyrimas) nustatyta, kad esant apibrėžtoms tyrimo sąlygoms sistema reikalauja iki 204 kartų mažiau atminties. Tad pasirinkus šį metodą nereikės įsigyti didelio kiekio operatyvinės atminties, esant didžiulėms (daugiau nei 1000 užklausų vienu metu iš skirtingų vartotojų) IS apkrovoms.
5. Išanalizavus atsako laiko charakteristikų tyrimų (5.4.1, 5.4.3 ir 5.4.5 tyrimai) rezultatus galime teigti, kad visų trijų parametrų (vartotojų, funkcijų ir grupių skaičius) įtaka atsako laikui yra tiesiogiai proporcinga esant didelėms IS apkrovoms (daugiau nei 1000 užklausų vienu metu iš skirtingų vartotojų), tad išsimatavę savo įrangos našumo ( $k$ ) koeficientą galime nusakyti būsimos sistemos maksimalų vartotojų skaičių, prie kurio bus tenkinamas IS specifikacijoje nurodytas minimalus atsako laikas.
6. Pritaikius pasiūlytą teisių valdymo metodą LSMUSIS IS atitiko specifikacijoje aprašytą minimalų atsako laiką bei serveryje sunaudojo mažiau operatyvios atminties. Tad galime teigti, kad darbo metu suprojektuotas ir realizuotas metodas pasiteisino bei sutaupė serverio resursų.
7. Šio darbo metu sukurta programinė įranga yra įdiegta ir sėkmingai naudojama nuo 2013m rugsėjo mėnesio Lietuvos sveikatos mokslų universitete. Tai įrodantis aprobavimo aktas yra pateikiamas šio darbo priede.



## 7 LITERATŪROS SARAŠAS

- [1] David Elliott Bell „Looking Back at the Bell-La Padula Model“ 7 Gruodis 2005. [Tinkle]. Available: <http://www.acsac.org/2005/papers/Bell.pdf> [Kreiptasi 13 Vasaris 2014]
- [2] Jago Maniscalchi „Information Security Models for Confidentiality and Integrity“ 17 Gegužė 2010. [Tinkle]. Available: <http://www.digitalthreat.net/2010/05/information-security-models-for-confidentiality-and-integrity/> [Kreiptasi 18 Vasaris 2014]
- [3] Joshua Brindle „Secure doesn't mean anything“ 30 Kovas 2008. [Tinkle]. Available: <http://securityblog.org/2008/03/30/secure-doesnt-mean-anything/> [Kreiptasi 10 Kovas 2014]
- [4] Jonathan Sander „RBAC and ABAC and Roles“ 3 Lapkritis 2009. [Tinkle]. Available: <http://identitysander.wordpress.com/2009/11/03/rbac-and-abac-and-roles-oh-my/> [Kreiptasi 12 Kovas 2014]
- [5] Lietuvos sveikatos mokslų universitetas „Lietuvos sveikatos mokslų universiteto pirmosios ir antrosios pakopų bei vientisųjų studijų reglamentas“ 7 Birželis 2013. [Tinkle]. Available: [http://ismuni.lt/media/dynamic/files/2108/lsmu\\_studiju\\_reglamentas\\_2013.pdf](http://ismuni.lt/media/dynamic/files/2108/lsmu_studiju_reglamentas_2013.pdf) [Kreiptasi 18 Kovas 2014]
- [6] Bob Tarzey „Extending Microsoft Active Directory“ 3 Sausis 2014. [Tinkle]. Available: <http://www.computerweekly.com/blogs/quocirca-insights/2014/01/extending-microsoft-active-dir.html> [Kreiptasi 20 Kovas 2014]
- [7] „StructureMap“ 2014. [Tinkle]. Available: <http://docs.structuremap.net/> [Kreiptasi 25 Kovas 2014]
- [8] Tom Dykstra „Implementing the Repository and Unit of Work Patterns in an ASP.NET MVC Application“ 30 Liepa 2013. [Tinkle]. Available: <http://www.asp.net/mvc/tutorials/getting-started-with-ef-5-using-mvc-4/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application> [Kreiptasi 18 Kovas 2014]
- [9] Microsoft „Caching Architecture Guide for .NET Framework Applications“ Balandis 2013. [Tinkle]. Available: <http://msdn.microsoft.com/en-us/library/ee817645.aspx> [Kreiptasi 29 Kovas 2014]
- [10] Scott Price „Web Performance Optimization“ 13 Liepa 2011. [Tinkle]. Available: <http://loadstorm.com/2011/07/web-application-performance-optimization-part-2-application-caching/> [Kreiptasi 29 Kovas 2014]
- [11] Malcolm Sheridan „ASP.NET MVC 3– Global and Conditional Filters“ 22 Rugsėjis 2010. [Tinkle]. Available: <http://www.dotnetcurry.com/showarticle.aspx?ID=578> [Kreiptasi 2 Balandis 2014]

- [12] Rahul Rajat Singh „Understanding Filters and Attributes in ASP.NET MVC“ 14 Balandis 2013. [Tinkle]. Available: <http://www.codeproject.com/Articles/577776/Filters-and-Attributes-in-ASPNET-MVC> [Kreiptasi 2 Balandis 2014]
- [13] Microsoft „Entity Framework Overview“ 2012. [Tinkle]. Available: [http://msdn.microsoft.com/en-us/library/vstudio/bb399567\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/bb399567(v=vs.100).aspx) [Kreiptasi 5 Balandis 2014]
- [14] Ricardo Peres „Differences Between NHibernate and Entity Framework“ 7 Birželis 2012. [Tinkle]. Available: <http://weblogs.asp.net/ricardoperes/archive/2012/06/07/differences-between-nhibernate-and-entity-framework.aspx> [Kreiptasi 6 Balandis 2014]

## 8 PRIEDAI



### LIETUVOS SVEIKATOS MOKSLŲ UNIVERSITETAS INFORMACINIŲ TECHNOLOGIJŲ CENTRAS

Kauno Technologijos universitetas  
Informatikos fakultetas  
Studentų 50- 411, LT 3031 Kaunas  
Tel. (8 37) 300350  
Faksas (3 37) 300352

Lietuvos sveikatos mokslų  
universitetas  
Informacinių technologijų centras  
Sukilėlių g. 17, LT50009 Kaunas  
Tel. (8 37) 794775

#### PROGRAMŲ SISTEMOS PERDAVIMO IR APROBAVIMO AKTAS

20 14 m. sausio 9 d.

Programų sistemos pavadinimas LSMU studijų informacinė sistema

Kūrinio tipas Programinė įranga

Programų sistemos sukūrimo data 20 13 m. Rugpjūčio 1 d.

Kūrinio įteikimo UŽSAKOVUI data 20 13 m. Rugsėjo 1 d.

Užsakovo arba trečiojo asmens Kūrinio aprobavimo rezultatas:

Programinė įranga įdiegta į universiteto serverį.

Kūrinio aprobavimo data 20 13 m. Spalio 1 d.

Kūrinio originalo saugotojas - LSMU informacinių technologijų centras

#### AUTORIAI

Evaldas Zimnickas

(vardas, pavardė)

(parašas)

Tomas Pozingis

(vardas, pavardė)

(parašas)

Martynas Rimkus

(vardas, pavardė)

(parašas)

#### UŽSAKOVAS

Kęstutis Šimatonis

(vardas, pavardė)

(parašas)

Asmens pareigų pavadinimas

LSMU ITC direktorius  
Kęstutis Šimatonis

(Parašas)

(Vardas ir pavardė)

Lietuvos sveikatos mokslų universitetas  
Informacinių technologijų centras  
Sukilėlių g. 17, LT50009 Kaunas  
Tel. (8 37) 794775