

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
VERSLO INFORMATIKOS KATEDRA

Edvinas Miliauskas

**Naujos generacijos intelektualių tinklų
orientuotų IMS architektūrai demonstracinės
paslaugos kūrimas**

Magistro darbas

Darbo vadovas

doc. dr. Vytautas Pilkauskas

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
VERSLO INFORMATIKOS KATEDRA

Edvinas Miliauskas

**Naujos generacijos intelektualių tinklų
orientuotų IMS architektūrai demonstracinės
paslaugos kūrimas**

Magistro darbas

Recenzentas

prof. dr. Rimantas Butleris
2010-05

Vadovas

doc. dr. Vytautas Pilkauskas
2010-05

Atliko

IFM-4/2 gr. stud.
Edvinas Miliauskas
2010-05-31

Kaunas, 2010

Development of IMS ready sample NGIN Services

SUMMARY

During course of this project we developed an telecommunications service - virtual private numbering. It allows users to create, manage and be accessible by short numbers. At first we analysed telecommunication field, then - possible platforms for SIP based service and chose open source JSLEE certified „Mobicents“. In next step, there were formulated requirements, designed architecture, developed and tested service. After that, we investigated possibilities to improve service execution parameters and it's current use of resources, influence to platform. Lastly investigation were checked by experiments.

Turinys

1. Įvadas.....	7
1.1. Dokumento paskirtis.....	7
1.2. Santrauka.....	7
2. NGIN paslaugos kūrimo analizė.....	8
2.1. Telekomunikacijos paslaugų sritis.....	8
2.1.1. Srities svarbos Lietuvoje įvertinimas.....	8
2.1.2. IMS architektūros aktualumas ir tikslingumas.....	10
2.2. Telekomunikacijų paslauga veikianti IMS architektūroje.....	13
2.2.1. Bendros charakteristikos ir reikalavimai.....	13
2.3. Paslaugos realizavimo problemos.....	15
2.3.1. Realizavimo platformos pasirinkimas.....	15
2.3.1.1. SIP servlets ir JSLEE palyginimas.....	16
2.3.1.2. JSLEE specifikacijos platformos.....	20
2.3.1.3. JSLEE specifikacijų versijos 1.0 ir 1.1.....	22
2.3.2. Priimtas sprendimas.....	23
3. Paslaugos realizacija.....	24
3.1. Reikalavimų specifikavimas.....	24
3.1.1. Sistemos paskirtis.....	24
3.1.1.1. Kūrimo pagrindas.....	24
3.1.1.2. Sistemos tikslai.....	25
3.1.2. Vartotojai.....	25
3.1.3. Įpareigojantys apribojimai.....	27
3.1.3.1. Apribojimai sprendimui.....	27
3.1.3.2. Diegimo aplinkai.....	27
3.1.4. Svarbūs faktai ir prielaidos.....	27
3.1.5. Veiklos sfera.....	28
3.1.5.1. Veiklos kontekstas.....	28
3.1.5.2. Bendradarbiaujančios sistemos.....	29
3.1.5.3. Veiklos padalinimas.....	29
3.1.6. Produkto veiklos sfera.....	30
3.1.6.1. Sistemos ribos.....	30
3.1.7. Funkciniai reikalavimai.....	31
3.1.8. Nefunkciniai reikalavimai.....	31
3.1.9. Naujos problemos.....	32
3.1.10. Rizikos.....	33
3.1.10.1. Galimos sistemos kūrimo rizikos.....	33
3.1.10.2. Atsitiktinumų valdymo planas.....	33
3.1.11. Vartotojo dokumentacija ir apmokymas.....	33
3.1.12. Perspektyviniai reikalavimai.....	34
3.2. Architektūros specifikacija.....	34
3.2.1. Architektūros pateikimas.....	34
3.2.2. Architektūros tikslai ir apribojimai.....	35
3.3. Sistemos statinis vaizdas.....	35
3.3.1. Apžvalga.....	35
3.3.2. Paketų detalizavimas.....	36
3.4. Sistemos dinaminis vaizdas.....	37
3.5. Išdėstymo vaizdas.....	41
3.6. Duomenų vaizdas.....	43
3.7. Kokybė.....	43
4. Paslaugos tobulinimo galimybės.....	45

4.1.Naujos JSLEE versijos įvertinimas.....	46
4.2.Naujos „Mobicents“ versijos įvertinimas.....	47
4.3.Duomenų kiekio įtakos įvertinimas.....	49
5.Eksperimentinė dalis.....	51
5.1.Matavimo aplinka.....	51
5.2.Paslaugos įtakos platformai įvertinimas.....	52
5.3.Platformos versijos pakeitimo įvertinimas.....	53
5.4.Duomenų bazės įtakos paslaugos našumui įvertinimas.....	57
6.Išvados.....	59
7.Literatūra.....	61
8.Terminų ir santrumpų žodynas.....	63
9.Priedai.....	65
9.1.Matavimo aplinka, įrankiai bei konfigūracija.....	65

Lentelių turinys

1 lentelė: „Open cloud“ SIP Servlet ir JSLEE palyginimas.....	18
2 lentelė: JSLEE specifikavimo grupės JSLEE ir SIP Servlets palyginimas.....	19
3 lentelė: Veiklos įvykių sąrašas.....	29
4 lentelė: Galimos rizikos ir jų įvertinimas.....	33
5 lentelė: Galimas VPN duomenų bazės užpildymas.....	50

Paveikslėlių turinys

1 pav. Fiksuoto ir judriojo ryšio abonentų skaičius, tūkst.....	9
2 pav. Tinklai, prie kurių jungiasi IMS.....	11
3 pav. JSLEE specifikacijos architektūra.....	23
4 pav. Konteksto diagrama.....	28
5 pav. Panaudojimo atvejų diagrama.....	30
6 pav. Sistemos paketai.....	35
7 pav. Sekų diagrama kai paslauga suteikiama be nesklandumų.....	37
8 pav. Bendradarbiavimo diagrama, kai paslauga nesuteikiama dėl negaliojančio trumpo numerio.....	37
9 pav. Paslaugos teikimo būsenų diagrama.....	38
10 pav. Sistemos konfigūracijos valdymo būsenų diagrama.....	38
11 pav. Sistemos administravimo veiklos diagrama.....	39
12 pav. Paslaugos teikimo veiklos diagrama.....	40
13 pav. Išdėstymo vaizdas.....	41
14 pav. JSLEE specifikacijos architektūra.....	42
15 pav. Sistemos naudojamos duomenų bazės diagrama.....	43
16 pav. 1.0 JSLEE platformos atsakymo laikas.....	52
17 pav. Platformoje apdorojamų skambučių keikis per sekundę.....	53
18 pav. Blogų žinučių kiekis pagal platformą.....	54
19 pav. Pakartotinis įvykių siuntimas pagal platformą.....	55
20 pav. Padarytų pakartotinių siuntimų palyginimas naudojant skirtingus GC.....	56
21 pav. Procesoriaus sąnaudų įvertinimas su skirtingo tipo GC.....	56
22 pav. 1.0 versijos VPN atsakymo laiko santykis pagal duomenų bazės pilnumą.....	58
23 pav. Procesoriaus užimtumas pagal duomenų bazės užpildymą.....	58

1. ĮVADAS

1.1. Dokumento paskirtis

Magistro darbe pristatomas demonstracinės telekomunikacijų paslaugos kūrimas, su tuo susijusios problemos ir jų sprendimas. Tai keturių KTU programų inžinerijos magistro studijų semestrų darbas. Pagrindiniai dokumento skyriai: analizė, paslaugos realizacija, tyrimas bei eksperimentas.

Analizėje pristatoma tyrimo sritis, jos svarba, tyrimo objektas bei problemos.

Projektinėje dalyje supažindinama su realizuota programa – pirmų trijų magistro semestrų darbas. Čia dokumentuojami reikalavimai, sistemos architektūra bei kontekstas.

Tyrimo dalyje bandoma įvertinti, ką naudinga patobulinti, kad pagerinti norimas charakteristikas. Šiame skyriuje pristatoma informacija, kuria remiantis vėliau atliekami eksperimentai priimtų prielaidų patvirtinimui, paneigimui ar naujų atradimų padarymui

Eksperimentinėje dalyje atliekami teorinių prielaidų patvirtinimas ar paneigimas praktinėmis priemonėmis.

1.2. Santrauka

Internetui tampa vis populiarešniam, jo panaudojimo sritys plečiasi, ko pasekoje keičiasi technologijos, atsiranda naujos galimybės... Ne išimtis ir telekomunikacijos. Ilgą laiką telefonų paslaugos apsiribojo tik dviejų vartotojų sujungimu pokalbiui. Vėliau atsirado kitos skambučių valdymo paslaugos (paskutinio skambinusio rinkimas, skambučio nukreipimas, blokavimas...), tačiau jos buvo įgyvendinamos firminėje įrangoje taigi operatoriai turėjo naudoti vienos įrangos tiekėjus jei norėjo teikti geros kokybės paslaugas. Sekantis žingsnis buvo paslaugų atskyrimas nuo šios firminės įrangos ir jos patalpinimas atskiruose terminaluose.

Šiuo metu bandoma perkelti duomenų srautus į internetinį paketų siuntimą. Interneto protokolo naudojimas iš dalies mažina kaštus bei operatorių pelną, tačiau jo pagalba galima suteikti daugybę įvairių naujų paslaugų taip sukuriant naujas ar praplečiant esamas rinkas.

Darbo metu buvo susipažinta su telekomunikacijų sritimi, realizuota paslauga, atliktas jos tobulinimas ir jo įvertinimas. Pagrindinės nagrinėjamos problemos yra susijusios su šios paslaugos realizavimu bei aplinkos pasirinkimu.

2. NGIN PASLAUGOS KŪRIMO ANALIZĖ

2.1. Telekomunikacijos paslaugų sritis

Darbo tikslas yra susipažinti su šiuolaikinėmis telekomunikacijų paslaugomis, jų kūrimo priemonėmis. Konkrečiau IMS architektūroje realizuoti demonstracinę NGIN paslaugą besinaudojančią SIP signalizavimo protokolu. Paslaugos pobūdis priklauso nuo programuotojo (autorius), tačiau rekomenduojama realizuoti vieną ar kelias iš standartinių GSM tinklo paslaugų.

Sukurtos programos galutinis adresatas bus vartotojas besinaudojantis IMS architektūros telefoniniu tinklu. Paslauga reguliuos skambučio būsenas: nukreips, uždraus, sulaukys...

Kadangi nebus realizuojama sudėtinga paslauga, ji nebus platinama kaip atskiras produktas. Perspektyvūs klientai pageidauja viso paketo: paslaugų, jų platformos bei aparatinės įrangos.

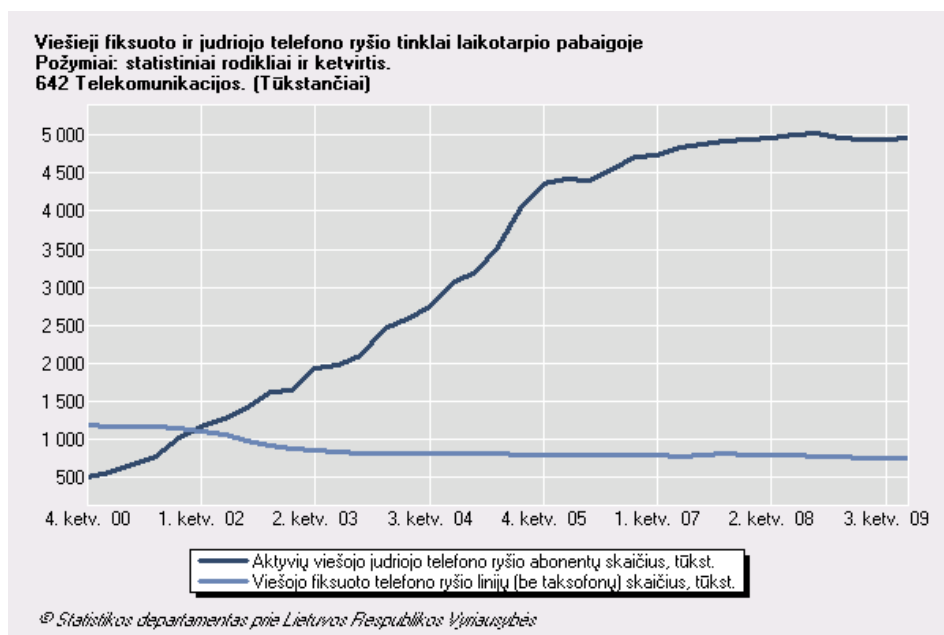
Sukurtos sistemos/paslaugos galimi pirkėjai yra operatoriai arba didelės kompanijos, valstybinės įstaigos, norinčios įsigyti ar patobulinti nuosavą telekomunikacijų sistemą. Kaip žinome didelėms kompanijoms vienas svarbiausių reikalavimų yra kokybės užtikrinimas, pastovumas ir palaikymas. Dažniausiai kompanijos laukia pasiūlymų ir perka pilną komplektą: visą platformą su paslaugomis, serveriais, palaikymu bei kita.

2.1.1. Srities svarbos Lietuvoje įvertinimas

Lietuvos respublikos ryšių reguliavimo tarnybos duomenimis [1] Lietuvoje veikia 10 fiksuoto ryšio operatorių: 7 iš jų naudojami IP tinklu duomenų perdavimui. UAB TEO teikia fiksuoto ryšio paslaugas tiek naudodamasi IP duomenų perdavimui, tiek standartiniais tinklais. Toks IP paremtų operatorių skaičius yra dėl to, kad šios paslaugos teikimas reikalauja tik interneto ryšio ir kiekvienas interneto ryšio tiekėjas gali be didesnių rūpesčių ją teikti. Ne interneto ryšio teikėjams gali kilti problemų dėl paketų perdavimo (svarba, QoS...), tačiau tai yra išsprendžiama, kad ir sunkiau negu savo tinkle teikiantiems paslaugas interneto tiekėjams.

Tuo tarpu judriojo ryšio paslaugą, pagal tą patį šaltinį [1], teikia 7 operatoriai. Čia minimi tik GSM tinklo operatoriai. Judriajam ryšiui perduoti galima naudoti ir kitus tinklus: 3G (o tai pat ir senesnius ar naujesnius interneto perdavimo protokolus GSM tinklu), WiFi, WiMAX. Pasinaudojant šiais tinklais bei tam pritaikytais telefonais, vartotojai gali tiesiogiai naudotis internetu, o kartu ir telefonija, prieinama fiksuotojo ryšio naudotojais ar paprastais internetu vartotojams (šiuo atveju turima omenyje Skype, Frype, gTalk bei kitą bendravimui

internetu skirtą programinę įrangą). Tačiau šis metodas (judrusis ryšys ne GSM tinklu) nėra populiarus, nes alternatyvių tinklų apimties zona nėra tokia didelė ar teikiamas nepakankamos kokybės interneto ryšys. Be to, dažnai reikalinga papildoma programinė ar techninė įranga, yra reikalavimas perprasti naują technologiją.



1 pav. Fiksuoto ir judriojo ryšio abonentių skaičius, tūkst

Nepaisant visa to, vartotojų naudojančių įvairius ryšio tinklus bus. Migracija buvo iš fiksuoto link judriojo, labai tikėtina, kad ateityje tokia migracija įvyks iš GSM tinklo link internetinio. Tačiau kaip matome iš 1 pav pavaizduotos diagramos [2], fiksuoto telefono abonentių skaičius mažėja, bet ne taip staigiai. Taigi, artimiausią dešimtmetį Lietuvoje tikrai bus vartotojų besinaudojančių apibūdintais trim tinklais (fiksuotas, GSM, VoIP). Šiuos tinklus apjungia IMS architektūra ir jiems gali būti teikiamos vienodos paslaugos.

Operatoriai stengiasi sudominti, patenkinti klientus pateikiant jiems patogias, naujas funkcijas. Šiuo metu veikia nemažai internetą naudojančių telefonu prieinamų paslaugų: video pokalbiai, skambinimas į Skype ir panašias internetines bendravimo aplikacijas, vietos nustatymas bei kita. Taigi turint kokybišką ir perspektyvią paslaugos idėją, ją ar jos realizaciją parduoti Lietuvos rinkoje yra įmanoma, o verslo sėkmė priklauso nuo idėjos turinio ir priimtimumo vartotojams.

Tuo tarpu pilno paketo pardavimas (paslaugos su architektūra) Lietuvoje yra ribotos. Kažkaip konkrečiai įvertinti padėtį Lietuvoje neišeis. Galime tik padaryti išvadą iš to, jog Lietuvoje veikia sąlygiškai nedidelės kompanijos, kurioms nuosavas telekomunikacijų tinklas nėra vienas iš prioritetų.

2.1.2. IMS architektūros aktualumas ir tikslumas

Telefoninės paslaugos ilgai liko nepakitusios. Tačiau laikui bėgant, telefoninių paslaugų teikėjai suprato, jog neužteks sujungti klientą A su klientu B. Jie gali būti užprogramuoti persiųsti, blokuoti, perkelti skambučius, įrašyti garso žinutes... Kitaip tariant, jie galėjo atlikti viską ką leidžia instaliuota aparatinė ir programinė įranga. [3]

Tuo tarpu pastaraisiais metais senesniems telefoninių paslaugų teikėjams išnykimu grasina sparčiai plintanti paslauga – Internetas.

Interneto protokolas yra visa apimantis. Pagal Pasaulinės interneto statistikos 2008 metų gegužės 30 dienos duomenis, pasaulyje yra netoli 1,5 milijardo interneto vartotojų, kas sudaro 300% padidėjimą 2000-2008 metų laikotarpyje [4]. Internetas leidžia žmonėms bendrauti naudojant skirtingus terminalus, o toks paplitimas ir plati pritaikymo sritis suteikia įvairias verslo galimybes, kurias ir bandoma realizuoti telekomunikacijų aplinkoje.

Nors pirmosios kartos internetas buvo daugiausiai skirtas perduoti ne realaus laiko duomenims, pastaraisiais metais paslaugos su griežtais paslaugų kokybės reikalavimais (angl. QoS – *Quality of Service*) vis labiau plinta šiame duomenų perdavimo protokole. Be to, operatorių pelnas, gaunamas iš multimedijos paslaugų, didėja [5]. Interneto protokolo (IP) architektūros panaudojimas visų paslaugų teikimui tampa vis stipresne tendencija.

Taigi didėjanti Google, Skype, Yahoo ir panašių internetu paremtų komunikacijos teikėjų sėkmė yra rimta grėsmė tradiciniams operatoriams ir nuo jų sėkmės priklausančių prekyautojų bazei. Dėl keleto priežasčių mobilūs įrenginiai suteikia prastos kokybės internetinės telefonijos (VoIP) paslaugas, tačiau tai keisis kelių metų bėgyje.

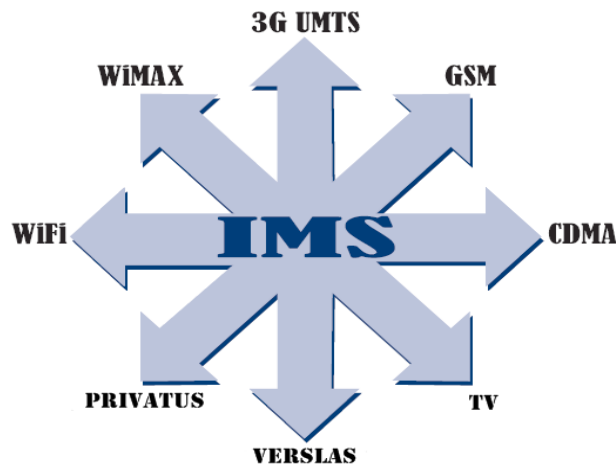
Štai keletas techninių, ekonominių priežasčių, dėl kurių VoIP yra geriau už tradicinę, laidinę telefoniją [6]:

- Daug lengviau instaliuoti ir konfigūruoti už firminę telefoninę sistemą
- Lengvesnė priežiūra internetinio puslapio ar kitos vartotojo sąsajos dėka
- Žymus kaštų sumažimas naudojant VoIP paslaugos teikėjus
- Telefoninių laidų pašalinimas
- Nepriklausomybė nuo pardavėjo – galima naudoti įvairių gamintojų telefonus su bet kuriuo SIP paremtu PBX, VoIP paslaugos teikėju...
- Lengvai praplečiama
- Geresnis vartotojų aptarnavimas ir produktyvumas

- Daugiau telefonų sistemos savybių už mažesnę kainą
- Lengvas telefono vietos keitimas
- Lengvesnis telefonų naudojimas – daug lengviau nei firminėse sistemose įvykdomos sudėtingesnės telefoninės funkcijos.

Telefoninių paslaugų teikėjai negali ilgiau sau leisti siūlyti tik balso paslaugas ir vis greitesnę interneto prieigą, kadangi tai tapo įprastinėmis prekėmis. Tiekėjai turi atrinkti patrauklias paslaugas, kurios gali būti pateikiamos su baziniu prieigos teikimu.

Mobilių telefonų tinklų operatoriai pasiūlė portalus (CDMA, i-mode, GPRS) tam, kad būtų galima teikti kitokio turinio paslaugas mobiliems telefonams. Šie bandymai pasibaigė skirtingai: i-mode yra populiarus, tačiau WAP neprisigijo Europoje. [7]



2 pav. Tinklai, prie kurių jungiasi IMS

Mobilių telefoninių paslaugų teikėjai negali pasiūlyti multimedijos komponentų (teksto, nuotraukų, video, garso) apjungimo laisvės vieno skambučio metu. Dvipusis telefoninis pokalbis negali būti praplečiamas iki grupinio garso ir vaizdo pašnekesio. IMS įveikia šį apribojimą. Dėl IP pamatų ir SIP (IMS signalizavimo protokolas) plečiamumo IMS leidžia maišyti duomenų ir telekomunikacijos paslaugas. [7]

IMS – tai IP multimedijos posistemis. IMS sudaro IP paremtas tinklas, kuris jungiasi prie įvairių prieigos taškų suteikiantis vieną paslaugą bevielio, linijinio ir kabelinio tinklo vartotojams. Iš pradžių IMS modelis buvo sukurtas išplėsti GSM mobiliųjų tinklų galimybes, tačiau vėliau buvo pridėtos galimybės palaikyti kitus tinklus nei GPRS. Pavyzdžiui bevielis tinklas (WLAN), CDMA2000 bei laidinis tinklas.

IMS suteikia lanksčią IP multimedijos valdymo ir sesijų kontrolės platformą, kurią operatoriai gali apdengti esamą tinklo infrastruktūrą. Kaip perdanga, IMS operatoriams leidžia išpirkti ilgalaikę esamo tinklo įrangos vertę, sumažinant reikalaujamas kapitalo

investicijas kuriant ir diegiant naują paslaugą. Per IMS, prieiga prie tinklo paslaugų gali būti užtikrinta per interneto sąsają, leidžiančią trečiųjų šalių kūrėjams, paslaugų teikėjams ir netgi abonentams savarankiškai tvarkyti savo paslaugų patirtį, o tinklo operatorius pasilieka galimybę kontroliuoti tinklo išteklius. Supaprastinta, saugi prieiga visoms šalims, reiškia mažiau tinklo personalo išteklių naujų paslaugų valdymui, o tai savo ruožtu mažina tiekimo ir operacijų išlaidas ir suteikia didesnę kainų lankstumą.

Su IMS tinkle, abonentai galėtų kontroliuoti, kada ir kaip jie bendrauja. Jie gali pasirinkti labiausiai tinkamą terpę arba jų derinį, medijos ir vaizdo, balso, teksto, atvaizdus, arba momentinius pranešimus, visus turėti vienu metu ir realiame laike.

Remiantis IMS, operatoriai gali į rinką greitai įdiegti naujas paslaugas, o taikydami į naujas rinkas - pritraukti naujų pajamų srautų. Trečiosios šalies ištekliai aplikacijų kūrimui ir priežiūrai gali būti valdomi efektyviau ir saugiau, pateikiant aibę sudėtingų paslaugų, skirtų pritraukti stambius klientus.

IMS naudoja sesijos inicializacijos protokolą (SIP) skambučių kontrolei, kurios metu skambutis/sesija yra perduodama tarp skirtingų skambučio valdymo elementų. Kadangi SIP yra paprastas protokolas ir nėra specialiai sukurtas jį naudojančioms aplikacijų funkcionalumui patenkinti, jis tampa geru metodu skambučio kontrolei, ypač kai ateities paslaugos dar nėra numatytos.

IMS centrinis tinklas yra apibrėžiamas kaip sluoksninis tinklas, susidedantis iš multimedijos transportavimo sluoksnio (garsas, video, duomenys...), skambučių kontrolės sluoksnio (SIP kontrolė, perdavimo kontrolė...) ir programų sluoksnis (suteikiantis papildomą funkcionalumą IMS centriniam tinklui). IMS centrinio tinklo naudingiausia savybė yra bendro programų sluoksnio naudojimas. Bendri transporto ir skambučio kontrolės sluoksniai vartotojams suteikia įvairias paslaugas tarp skirtingo prisijungimo tinklų. Tai leidžia operatoriams savo klientams pasiūlyti vienodas paslaugas skirtinguose tinkluose. Kaip jau minėta anksčiau, sukurtos naujos aplikacijos gali būti lengvai įdiegiamos su mažesniais kaštais panaudojant esamus tinklo elementus ir prisijungimo taškus.

Tiriamajame darbe bus labiau domimasi ir bandoma įgyvendinti standartines telekomunikacijos paslaugas. Šios paslaugos yra jau kurį laiką realizuotos sumaniajame tinkle (IN – angl. *intelligent network*). IN – nepriklausomų paslaugų telekomunikacijų tinklas. Tiksliau, sumanumas yra išimtas iš jungiklio (aparatinės telekomunikacijų tinklo dalies) ir perdėtas į kompiuterinius mazgus, kurie yra paskirstyti visame tinkle [8]. Šiuo metu IN paremta tinklo architektūra yra atnaujinama į NGIN (IMS ar bendriau – IP) paremtą

architektūrą. Tačiau nors ir tinklo architektūra kinta, vartotojams teikiamos paslaugos turi nemažėti. Taigi migracijos tarp architektūrų metu, senosios paslaugos turi būti perdarytos taip, jog jomis galima būtų naudotis ir naujos architektūros tinkle.

IN paslaugų pavyzdžiai [8,9]:

- Trečios šalies skambutis
- Skambučio pranešimas
- Skambučio valdymas
- Konferencija
- Netrukdyti (DND)
- Skambučio atranka
- Telefono numerio mobilumas
- Nemokamas skambutis
- Skambučių paskirstymas remiantis įvairiais skambučio kriterijais (vietovės, laiko, proporcinis paskirstymas)
- Išankstinis apmokėjimas
- ir t. t.

IMS paremtam tinkle veikiančių paslaugų įvairovė priklauso tik nuo kūrėjų, užsakovų vaizduotės. Tačiau daliai jų įgyvendinti kol kas gali sutrukdyti įrankių stoka ar jų netobulumas, rinkos nebuvimas, sudėtingas ir ilgas kūrimo procesas bei kiti veiksniai. Susipažinimui su aplinka, joje kylančiomis problemomis, jų įtakojančių kriterijų svarba, pakanka realizuoti jau žinomas, netokias sudėtingas paslaugas.

2.2. Telekomunikacijų paslauga veikianči IMS architektūroje

2.2.1. Bendros charakteristikos ir reikalavimai

Dvi iš esmės prieštaraujančios kryptys įtakoja IMS architektūrą. Šioms kryptims keliami reikalavimai bei jų svarba skiriasi [10] :

- Plačioji rinka – poreikis teikti paslaugas atitinkančias globalinį pasiekiamumą ir poveikį atitinkančius tradicinę telekomunikaciją. Tai yra standartizuotos paslaugos, palaikomos plataus rato terminalų ir veikiančios tarptautinėje operatorių bendruomenėje. Jas charakterizuoja: plečiamumas, eksploatacinė parengtis ir eksploatacinės savybės, o funkcinis pokytis nusakomas standartų.

- Nestandardizuotos individualiems operatoriams siūlomos paslaugos skirtos vietinei rinkai, grupei operatorių ar netgi pasaulinei rinkai. Šios paslaugos remiasi IP bendruomenės kūrybiškumu ir dinamika. Jas charakterizuoja lankstumas ir greitas pateikimas rinkai.

Migruojant tarp architektūrų, joms taikomi reikalavimai turi būti tenkinami.

Telefoninėse paslaugose reikalavimus galime suskirstyti į šias grupes [11]:

- Paslaugų reikalavimai. Šie reikalavimai yra pateikiami infrastruktūrai skaidriam paslaugų pristatymui, grindžiami paveldo (pavyzdžiui, telefonijos) arba bendri reikalavimai, konkrečiai paslaugai. Šie reikalavimai gali skirtis tarp paslaugų teikėjų, ir galbūt net tarp skirtingų lygmenų panašiose paslaugose.
- Tinklo reikalavimai. Šie reikalavimai yra susiję su tinklu ir gali būti siejami su konkrečiomis paslaugomis arba būti pagrįsti bendrais IP NGIN infrastruktūros veikimo tikslais.
- Saugumo reikalavimai. Susiliejusiam tinkle yra žymiai griežtesni saugumo reikalavimai. Konkretūs reikalavimai prieigos prie paslaugų arba paslaugų mazgų apribojimui turi būti pateikta kaip sudėtinė infrastruktūros dalis ir saugumo politika.
- Veikimo reikalavimai. IP NGIN aplinka reikalauja skirtingų laiko stebėjimo priemonių nei tradicinė IP aplinka.
- Tinklo modelis. Visos anksčiau paminėtos reikalavimų grupės priklauso nuo tinklo modelio. Dauguma modelio sprendimų yra priimami remiantis sąnaudų-naudos analize. Svarbu tai, kad šie sprendimai būtų daromi sąmoningai.

Dauguma šių reikalavimų grupių nėra prieinamos, keičiamos realizuojant paslaugas, tačiau jeigu jos nebus tenkinamos yra didelė tikimybė, jog kuriama paslauga netaps sėkminga. Tik užtikrinus jog visi reikalavimai tenkinami – tinklas veiks kaip buvo numatyta ir atitiks operatoriaus (komercinį) lygį.

Kuriant paslaugą reikėtų atkreipti dėmesį į šiuos kriterijus:

- Paslaugos kokybė (QoS) – tai ryšio užtikrinimas. Teikiant balso ar vaizdo paslaugas didesni duomenų perdavimo trukdžiai yra greitai pastebimi vartotojo.
- Paslaugos yra teikiamos realiu laiku, taigi reikalaujami skaičiavimai, operacijos atlikimo laikas bet kuriuo galimu atveju neturi užtrukti ilgiau joms nustatyto laiko ar kitaip paveikti paslaugos kokybę.

- Patikimumas – paslauga turi būti teikiama esant visoms sąlygomis (be abejo čia neatsižvelgiama į aparatinius gedimus ar ryšio sutrinkimus tinkle). Turi būti atsižvelgta į galimas trukdžių priežastis (vartotojų srauto padidėjimas, atakos, vartotojų klaidos...) ir stengiamasi jas pašalinti ar kuo labiau sumažinti jų poveikį (pavyzdžiui, neaparnauti klientų kol serveris yra pilnai užkrautas).
- Saugumas – kaip ir kiekvienoje internetinėje programoje, turi būti užtikrinta, jog pašaliniai asmenys negalėtų prieiti ar gauti priėjimo prie jiems nepriklausomų duomenų bei operacijų.
- Aptarnaujančių vartotojų skaičiaus didinimas. Paslaugos atliekami skaičiavimai neturėtų reikalauti didelių aparatinių resursų. Didėjant vartotojų skaičiui paslaugos teikimo kokybė turi neprastėti, didėjimas turi būti numatytas ir galimas, o padidėjimo iššaukiamos išlaidos turi būti kuo mažesnės.

2.3.Paslaugos realizavimo problemos

2.3.1. Realizavimo platformos pasirinkimas

Plintat internetui, didėjat telefonijos klientų poreikiams, atsirado poreikis telefonų komunikacijose įtraukti ir internetą. Tam, kad atlikus minimalias modifikacijas esamai architektūrai ji galėtų naudotis interneto teikiama privalumais buvo sukurta IMS architektūra. Pagrindiniai IMS privalumai [12]:

- Integracija su esamomis architektūromis – IMS leidžia apjungti GSM, VoIP, standartinės bei kitus telefonijos tinklus.
- Duomenys, kur galima, perduodami interneto protokolu (IP), pvz SIP, tuo pačiu suteikiant galimybę teikti internetines paslaugas telefonų tinkle, pvz video telefonija, žaidimai tinkle ir kita.

Sesijos pražios protokolas (SIP) – tai protokolas, skirtas valdyti multimedijos komunikacijų sesijas. Šis protokolas bendrauja su keletu kitų protokolų ir yra atsakingas tik už komunikacijų sesijos signalizavimo dalį. Pagrindinės SIP protokolo realizavimo technologijos yra [13]:

- SIP Servlet – ji apibrėžia talpa paremtą modelį, kuris yra plačiai naudojamo Servlet modelio plėtinys. Ši technologija buvo suprojektuota supaprastinti SIP paremtų aplikacijų kūrimą taip paspartinant SIP protokolo įsisavinimą.

- JAIN SLEE arba JSLEE – apibūdina programų aplinką tinkamą tinklu paremtoms aplikacijoms. Čia apibrėžiamas nuo protokolų nepriklausomas, komponentinis ir objektiškai orientuotas programavimo modelis.

Dar prie technologijų yra priskiriama OSA/Parlay, tačiau ją galima naudoti kartu su JSLEE taip papildant vienas kitą. Taigi apibendrinus ir supaprastinus galime teigti, jog SIP Servlet yra skirtas susipažinimui, yra lengviau perprantamas, tačiau teikia mažiau galimybių už JSLEE technologiją.

JSLEE specifikaciją atitinkantys produktai yra:

- „Mobicents“ - atviro kodo, priklauso Red Hat
- „Amdocs jNetX“ – komercinis produktas, gamintojas jNetX
- „Rhino“ – komercinis produktas, gamintojas Open Cloud

Paminėtų ar kitų produktų pagalba yra kuriamos telekomunikacijų paslaugos, kurios veikia IMS architektūroje. Vienas tokių paslaugų kūrimo būdas yra užsakovo pageidavimų vykdymas, kai kuriama tik paslauga. Tuomet dirbama su kliento turima architektūra, realiais duomenimis todėl tokio tipo užsakymus atlieka su klientu glaudžius ryšius turinčios įmonės. Galutinis užsakymo produktas – paslauga, atliekanti kliento pageidaujamus veiksmus ir veikianti jo norimoje aplinkoje. Tokie užsakymai dažniausiai atkeliauja iš operatorių, kurie nori patobulinti ar papildyti jų klientams teikiamų paslaugų asortimentą.

Kitu atveju – klientas perka bendrą paketą: serverius, paslaugas, palaikymą, apmokymą, tobulinimą, pajungimą bei kita, pagal sutarties keliamus reikalavimus. Čia klientas dažniausiai būna didelės kompanijos, norinčios turėti nuosavą telekomunikacijų tinklą, telekomunikacijos paslaugas teikiančios įmonės (pvz. Telefonu teikiama informacija, telefoninės pagalbos centrai, kita). Tokie užsakymai atneša didesnę pelną, bet tuo pačiu reikalauja didesnės atsakomybės.

2.3.1.1. SIP servlets ir JSLEE palyginimas

Java paslaugų vykdymo aplinka (JSLEE – Java/JAIN Service Execution Environment) yra pasirinktas realizavimo standartas. Kitas realizavimo būdas galėtų būti SIP Servlets. Telekomunikacijų kompanijos dažnai turi apsispręsti tarp šių dviejų technologijų. Kiekvieno iš jų naudingumas gali keistis priklausant nuo projekto. Galimas JSLEE apibrėžimas [14]:

„JAIN SLEE apibrėžia aplikacijų platformą tinklui orientuotoms programoms. Ji apibrėžia nuo protokolų nepriklausantį, komponentinį ir objektinį ir tranzacinį programavimo modelį. JAIN SLEE yra pritaikytas naudoti konkretų resursą pagal resursų

adapterį, todėl SLEE iš esmės yra praplečiamas. Pagal programavimo modelio pobūdį ir JAIN SLEE teikiamas paslaugas, jis yra pritaikytas pirmojo lygio operatorių reikmėms.“

Tuo tarpu SIP Servlet [14] :

„SIP Servlets apibrėžia konteineriu grįstą modelį, kuris yra plačiai žinomo Servlet modelio praplėtimas. Jis buvo sukurtas siekiant supaprastinti SIP pagrindu sukurtų programų plėtrą ir tokiu būdu padidinti SIP protokolo priėmimą.“

Kiekvienos komunikacijos infrastruktūros svarbus bruožas ir SIP Servlet tikslas yra standartizuoti platformą SIP paremtų paslaugų teikimui. Šiuo atveju platforma – tai Java programavimo interfeisas bei standartai apibrėžiantys aplikacijų diegimą ir duomenų perdavimą.

Supaprastinus skirtumus - JSLEE suteikia daugiau galimybių, tačiau SIP Servlet yra paremtas Servlet standartais, kurie yra dažnai naudojami internetinių paslaugų kūrimo ir yra daug lengviau perprantami.

Toliau pateiktoje 1 lentelėje yra „Open Cloud“ įmonės suformuluotas SIP Servlet ir JSLEE palyginimas [15]. Ši įmonė teikia abu standartus atitinkančius produktus. Rinkoje nėra nusistovėjęs požiūris, kad kažkuris yra „geresnis“. Abu standartai yra pristatomi kaip papildantys vienas kitą, tačiau naudoti galima tik vieną, kurį turi nuspręsti atsakingi asmenys.

1 lentelė: „Open cloud“ SIP Servlet ir JSLEE palyginimas

	<i>Protokolai/Ypatybės</i>	<i>SIP Servlet</i>	<i>JSLEE</i>
<i>Susiję su IMS</i>	<i>(IMS reikalavimai)</i>		
	Palaikomas SIP	✓	✓
	Palaikomas daugiau nei vienas SIP API	✗	✓
	Įprasti SIP komponentai (pvz. proxy)	✓	✓
	Paslaugų bendravimas ir valdymas	✓	✓
	Palaiko kitus IMS protokolus (pvz. Diameter)	✗	✓
<i>OA&M</i>	<i>(labai svarbu)</i>		
	Standartinis sekimas, išpėjimai, naudojimas	✗	✓
	OA&M interfeisai (pvz. internetiniai atnaujinimai)	✗	✓
<i>Lankstumas</i>	<i>(plečiamumas ir migravimas į IMS)</i>		
	Ne IMS mazgų ir protokolų palaikymas	✗	✓
	Nereikalaujama privačių plėtinių	✗	✓
<i>Paslaugų kūrimas</i>	<i>(kokybė, patikimumas, pastangos, išlaidas, priežiūra)</i>		
	Geros programų architektūros skatinimas	✗	✓
	Yra nesėkmių modelis	✗	✓
	Atlieka sudėtingas programines užduotis	✗	✓
	Skatina pakartotinį panaudojimą	✗	✓

Sekantis palyginimas yra iš JSLEE specifikuojamos grupės [14]. Čia lyginamas iš standarto pusės apibrėžtas funkcionalumas. Dalis skirtumų gali būti suvienodinta pagalbinių bibliotekų, komponentų pagalba, tačiau tai yra nukrypimas nuo numatyto standarto ir gali neveikti ant kito gamintojo platformos. Iš šio palyginimo aiškiai matosi, kad SIP Servlets yra paprastesnis standartas su būtinomis priemonėmis ir savybėmis greitai ir lengvai susipažinti su SIP protokolu ir realizuoti jame veikiančią paslaugą.

2 lentelė: JSLEE specifikuojamos grupės JSLEE ir SIP Servlets palyginimas

SIP Servlet	JAIN SLEE
Aplikacijos architektūra: kaip aplikacijoje yra patalpinta biznio logika	
Paremti HTTP Servlet. Servlet yra aplikacijos vienetas. Nėra standartinio modelio kompozicijai ir pakartotiniam panaudojimui.	Paremtas komponentais, Objektinis modelis. Logikos vienetas yra SBB Kompozicijos ir pakartotinio panaudojimo palaikymas
Aplikacijos būseną: kaip aplikacijos būseną gali būti atvaizduota, pasiekama ir apsaugota	
Servlet'ai neturi būsenos Bendra būseną gali būti saugoma atskiram sesijos objekte kaip pora: pavadinimas reikšmė (Tekstas, Objektas) (<i>angl. String Object</i>) Bendra būseną yra prieinama visiems Servlet su priėjimu prie sesijos	SBB gali turėti būseną arba ne SBB būseną yra privati, aptinkama kompiliatoriaus (<i>angl. type safe</i>), perduodama ir yra pačio SBB kintamasis Bendra būseną gali būti saugoma atskirame veiklos kontekste (JSLEE komponentas) su kompiliatoriaus patikrinamu interfeisu Priėjimas prie būsenos gali būti nurodomas paleidimo metu
Lygiagrečumo kontrolė: kaip kontroliuojamos gijos	
Valdoma tik aplikacijos (pav. Java monitoriai)	Valdoma sistemos (pav. lygiagrečių tranzakcijų izoliacija)
Protokolų palaikymas	
SIP ir HTTP	Nepriklauso nuo protokolų Gali būti praplėstas palaikyti išorinius protokolus ir išorinius resursus Nuoseklus įvykių modelis, nepriklausantis nuo protokolo/resursų
Aplikacijoms prieinamos pagalbinės priemonės: platformos priemonės prieinamos joje veikiančiai aplikacijai	
Laikmatis	Laikmatis Žurnalas Žadintuvas Statistika ir naudojimas Profiliai
Prieinamumo mechanizmai: kaip savybės ir modelio koncepcijos gali būti išnaudojamos nuolatinio prieinamumo palaikymui	
Platformos valdoma būseną (sesijos objektas), kuri gali būti atkurta Jokio tranzakcijos konteksto SIP žinučių apdorojimui Netranzakcinės būsenos operacijos Neperduodamos pagalbinės priemonės Neapibrėžtas žlugimų apdorojimo modelis	Platformos valdoma būseną (SBB CMP...) Kuri gali būti atkurta Tranzakcijos kontekstas įvykio perdavimui Platformos valdomos būsenos operacijos gali būti perduodamos Pagalbinės priemonės gali būti perduodamos Apibrėžtas ir suprantamas žlugimo modelis tranzakcijų pagalba

Valdymas: platformos teikiamos priemonės aplikacijų ir pačios platformos valdymui	
Nėra apibrėžta jokių standartinių valdymo mechanizmų	Standartiniai valdymo interfeisai yra apibrėžti remiantis JMS (Java valdymo praplėtimams) Nepriklausomas nuo valdymo protokolo Interfeisas aplikacijų valdymui: gyvavimo ciklas, profiliai, žurnalas, atnaujinimai... SLEE gyvavimo ciklo valdymo interfeisas

Taigi apibendrinimui galime sutikti su JSLEE specifikavimo grupės išvadomis [14]:

„Jain SLEE specifikacija yra sudėtingesnė už SIP Servlets. JAIN SLEE yra standartizuotas didelio prieinamumo bei atsparumu gedimui mechanizmai programavimo modeliui ir aplikacijos gyvavimo ciklui. SIP Servlets yra programavimo modelis, JAIN SLEE yra programų platforma.“

2.3.1.2. JSLEE specifikacijos platformos

JSLEE yra specifikacija. Jos kūrimo procese dalyvauja atstovai iš įvairių su telekomunikacijomis, Java programavimu susijusių kompanijų. Pažymėtina, jog visų trijų JSLEE specifikaciją atitinkančių produktų gamintojai dalyvavo rengiant JSLEE 1.1 specifikaciją [17], tačiau JSLEE 1.0 kūrimo dalyvavo tik Open Cloud [16] atstovai. Pastarosios įmonės atstovas taip pat buvo vienas iš vadovaujančių visos specifikacijos kūrimui.

JSLEE specifikacijas atitinka trys produktai:

- „Amdocs jNetX“ – gamintojas jNetX (vėliau nupirktas Amdocs). Užsakovo naudojama JSLEE platformos realizacija.
- „Rhino“ – gamintojas Open Cloud.
- „Mobicents“ - priklauso Red Hat. Atviro kodo JSLEE specifikacijos platforma. Nors jos teikiamos galimybės dar neatitinka komercinių produktų, kūrimas yra aktyvus ir palaikomas rinkos dalyvių.

Sunku pateikti detalesnį šių platformų palyginimą. Pirmiausia dėl to, kad jos visos atitinka vienodą specifikaciją. Platesniam palyginimui reikia turėti gilų telekomunikacijų srities bei jos problemų suvokimą, būti pilnai išstudijavus JSLEE specifikaciją, turėti galimybę dirbti su visomis platformomis... Autorius neturi nei vienos iš šių galimybių bei žinių, todėl platformas gali įvertinti tik santykinai, pagal pateikiamus duomenis ir keliamus reikalavimus.

Sparta

Visi trys gamintojai pateikia savo platformos spartos įvertinimus, tačiau jie buvo vykdomi skirtingos aparatinės įrangos, veikiant skirtingoms paslaugoms. Tai neleidžia tiesiogiai jų įvertinti, tačiau sudaromas bendras išpūdis kas svarbu, į ką norima atkreipti dėmesį.

„Amdocs jNetX“ pateikia 2007 metais atliktą spartos įvertinimą [18]. Jis buvo atliekamas SUN T1 serveryje. Matavimo metu veikė VPN SS7 paremta paslauga, kurios metu buvo vykdomi keli kreipimaisi į duomenų bazę. Išvadose teigiama, tam kad neviršyti 60ms vėlavimo, reikėjo apdoroti ne daugiau kaip 600 skambučių per sekundę.

„Rhino“ kaip jNetX, pateikia taip pat 2007 atliktą spartos ir masto įvertinimą [19]. Šie testai buvo atliekami IBM BladeCenter-T HS20 serveriuose. Matavimo metu veikė Surask mane Sek mane (*angl. Find Me Follow Me*) paslauga. Rezultatuose nesimato, jog siekta sužinoti maksimalų skambučių per sekundę kiekį. Juose įrodinėjama kaip serverių resursų padidinimas vertikaliai (vieno serverio pagerinimas) ir horizontaliai (serverių kiekio padidinimas) stabiliai pagerina platformos darbą. Spartos atžvilgiu matyti, kad 300 skambučių yra apdorojami naudojant 10% procesoriaus. Sudėtingesnė paslauga sumažina apdorojančių skambučių kiekį iki 170 per sekundę.

„Mobicents“ 2010 metais atliko oficialų spartos įvertinimą [20]. Testas vykdytas Quad-Core AMD Opteron(tm) 2356 procesoriumi su 10Gb atminties. Matavimo metu veikė tik programa grąžinanti numatytus SIP atsakymus – SIP skambučio simuliacijai. Nustatyta, jog palaikoma 750 tokių skambučių per sekundę. Tai gana ryškus pagerėjimas nuo 2008 metų 185 skambučių per sekundę.

Kaip jau minėta, spartos rezultatų palyginti neina. Skiriasi viskas: matavimo metodai, aparatinė įranga, aplinka. Iš surinktos informacijos apie spartą galime spręsti, jog vienas serverinis kompiuteris palaiko ne mažiau kaip kelis šimtus skambučių per sekundę. Pagal Anandtech pateikiamą procesorių palyginimą [21] (Specjbb2005 testo rezultatai. Šis testas naudoja užsakymų apdoravimo aplikaciją parašytą Java kalba): „jNetX“ naudojamo Sun T1, „Rhino“ HS20 Intel Xeon bei „Mobicents“ 2x AMD Opteron sparta atitinkamai proporcinga kaip 2 : 1 : 1. Iš ko seka, jog „Mobicents“ testo rezultatai buvo pastebimai prastesni nepaisant paprastesnės matavimo programos, o komercinių produktų panašūs. Naujesnius matavimus pateikia tik „Mobicents“. Vėlgi šios vienintelės atviro kodo platformos rezultatus yra lengviausiai atgaminti – viskas yra pateikiama nemokamai ir su pilna dokumentacija. Tuo tarpu komercinių produktų rezultatai yra pasenę (~3 metų senumo) bei nepilnai dokumentuoti.

Įrankiai, dokumentacija, palaikymas

„jNetX“ internetiniame puslapyje daug informacijos nepavyko gauti. Kaip ir nemokamos demonstracinės platformos versijos. Konsultacijos vykdomos su individualiu klientu ar partneriu. Dokumentacija bei pavyzdžiai pateikiama kartu su produktu. Deja, tačiau pačio produkto ar su juo teikiamų įrankių nepavyko išbandyti. Kompanija įkurta 2001 metais.

„Rhino“ suteikia galimybę nemokamai išbandyti platformą. Internetiniame puslapyje pateikiama pilna dokumentacija, yra aktyvus forumas. Visa informacija yra viešai prieinama, informatyvi ir tvarkinga. Kompanija įkurta 2000 metais.

„Mobicents“ vienintelė atviro kodo JSLEE platformos implementacija. Didžiausias jos plusai – kaina bei laisvai prieinamas kodas. Projektą inicijavo Red Hat kompanija. Jos darbuotojai yra ir pagrindiniai projekto asmenys. Dokumentacija nėra pilna, kai kur pasenusi. Yra elektroninio pašto forumas su vis aktyvesne bendruomene.

2.3.1.3. JSLEE specifikacijų versijos 1.0 ir 1.1

Kaip jau minėta JSLEE turi dvi specifikacijos versijas:

- 1.0, išleista 2004 metais [16]. Visi trys produktai palaiko ir yra sertifikuoti šiai versijai.
- 1.1, išleista 2008 metais [17]. Tik Rhino 2.0 ir Mobicents 2.0 yra sertifikuoti šiai versijai. Joje gali veikti ir 1.0 specifikacijos komponentai.

Pagrindinės 1.1 specifikacijos versijos naujovės:

- Standartizuoti resursų adapteriai
- Keli svarbesni profilių pakeitimai
- Įtraukta nauja konteinerio koncepcija - bibliotekos. Dabar paslaugų aplikacijos turi labiau standartizuotą būdą dalintis bendromis klasėmis, bibliotekomis.
- Žadintuvai nuo dabar turi būseną su savo gyvavimo ciklu
- Praplėsta SBB CMP laukuose saugojimo objektų tipų aibė
- Patobulintas žurnalo pildymo komponentas.

Paslaugos kūrimo metu vienintelė „Rhino“ platforma buvo veikianti pagal 1.1 specifikaciją. „Mobicents“ savąjį išleido tik 2010 metais (jau sukūrus paslaugą), o jNetX dar nėra paskelbusi savo planų.

Naujos specifikacijos versijos naujovės neturi didelės įtakos realizuojamai paslaugai. Tačiau gali turėti skirtingus spartos rezultatus.

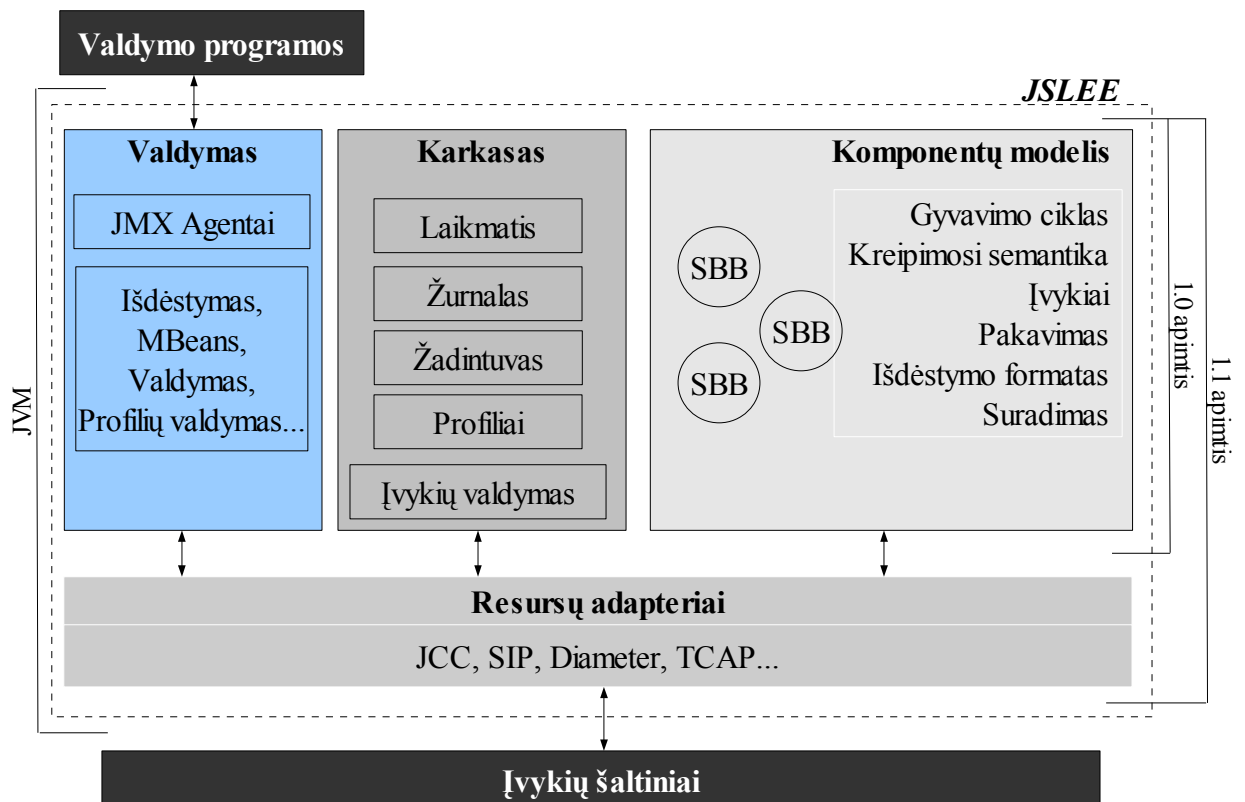
2.3.2. Priimtas sprendimas

Apsisprendžiant tarp JSLEE ir SIP Servlet pagrindinis faktorius buvo jų galimybės. Kol konkreti paslauga nebuvo nuo pradžių aiškiai apibrėžta, planuota jog ji turės palaikyti bendravimą su išoriniais resursais. Šiuo atveju SIP Servlet netiko, nes palaikomas bendravimas tik SIP protokolu ir su duomenų bazėmis, ko yra per mažai.

Reikalavimų rinkimo metu užsakovas išskyrė dvi JSLEE platformas: jNetX, kadangi yra jo naudojama, bei Mobicents, kadangi yra atviro kodo, perspektyvi alternatyva. Nuspręsta, kad paslauga bus kuriama Mobicents platformoje. Pasirinkimą įtakoję palankumai (Mobicents):

- atviras kodas – iškilus neaiškumams lengva rasti išsiaiškinti. Nėra prisirišimo prie vieno gamintojo. Galima laisvai modifikuoti.
- nemokama - darbo kontekste tai neturi ypač didelės reikšmės, tačiau svarbu užsakovui.
- sparčiai vystoma – projektas gauna vis didesnę palaikymą, gausėja atvira bendruomenė. Tai antra JSLEE 1.1 sertifikuota platforma.

Bendra platformos architektūra pateikta 3 pav. bei aprašyta trumpai po juo.



3 pav. JSLEE specifikacijos architektūra

Kaip matome iš 3 pav, JSLEE sudaro keturios pagrindinės dalys: valdymas, karkasas, resursų adapteriai ir komponentų modelis. 1.0 specifikacijoje resursų adapteriai buvo menkai

apibrėžti. 1.1 versijoje jie ir jų API yra pilnai specifikuoti. Visa JSLEE platforma veikia JVM. Resursai taip pat gali veikti JVM, tačiau to nereikalaujama.

Valdymo dalyje esantys komponentai leidžia valdyti visą JSLEE platformą.

Karkaso komponentai suteikia reikalingą funkcionalumą realizuotoms paslaugoms (SBB komponentai). Tarp jų svarbiausias – įvykių valdymo komponentas. Jis paskirsto gautus ir naujai sukurtus įvykius anksčiau prisiregistravusiems SBB ir resursams.

Resursų adapteriai sujungia komponentų modelį ir įvykių šaltinius (resursus). Įvykių šaltinis gali būti įgyvendintas bet kokia kalba ir aplinkoje. Resursų adapteris konvertuoja gaunamus specifinius tinklo įvykius ir protokolus į bendro pobūdžio, semantiškai lygiaverčius Java įvykius ir perduoda juos į JSLEE aplikacijų serverį tolesniam perdirbimui.

Mobicents veikia JBoss aplikacijų serveryje.

NGIN kontekste, „Mobicents“ yra pristatomas kaip aukštos spartos SDP bei IMS centrinis mazgas.

3. PASLAUGOS REALIZACIJA

3.1.Reikalavimų specifikavimas

3.1.1. Sistemos paskirtis

3.1.1.1. Kūrimo pagrindas

Galutinis produktas - telekomunikacijos paslauga veikianti IMS architektūroje ir realizuota JSLEE platformoje. Sukūrus produktą siekiam programuotoją supažindinti su telekomunikacijų programų kūrimu tam tikroje architektūroje bei naudojant konkrečias priemones. Bus realizuota virtualaus privataus numeravimo paslauga (*angl.* Virtual Private Numbering).

Virtualaus privataus numeravimo paslauga (toliau VPN), leidžia vartotojui susikurti trumpus numerius kuriais jis tampa pasiekiamas kitiems asmenims. Numatyta, kad šia paslauga pakol kas galės naudotis tik to pačio domeno SIP klientai. Jei bus skambinama į kitą domeną, prie trumpojo numerio reikės nurodyti ir patį domeną.

Galutiniam vartotojui VPN paslauga yra naudinga, nes:

- Jis pats gali reguliuoti savo trumpų numerių galiojimą taip turėdamas didesnę savo privatumo kontrolę. Tai yra, trumpi numeriai gali veikti kaip virtualus pašto adresai – gavus per daug šiukšlių/nutraukus veiklą, jį galima panaikinti ar perleisti kitam asmeniui.

- Lengviau atsiminti ir surinkti skambinant.
- Operatoriui tai patogiu, nes:
- Papildoma paslauga vartotojui.
 - Siūlomi įvairūs apmokestinimo būdai.

3.1.1.2. Sistemos tikslai

VPN paslauga tikrina jos veikiančiam domene vykstančius skambučius ir ieško trumpų numerių. Jei trumpas numeris yra aptiktas, skambučio duomenys pakeičiami, kad jis būtų nukreipiamas teisinga kryptimi. Skambučio vykdytojas pokyčio nemato ir yra automatiškai nukreipiamas į trumpo numerio savininko tikrąjį numerį.

VPN paslauga gali būti valdoma kliento. Sukurta sistema duoda priėjimą prie naudojamos duomenų bazės, kurią operatorius gali integruoti ar kreiptis iš savo paslaugų valdymo portalo/programinės įrangos.

Operatorius gali nurodyti kaip paslauga bus apmokestinama: mėnesinis mokestis, nemokama, vienetinis mokestis. Apmokestinimui naudojama operatoriaus apmokestinimo sistema. Į ją kreipiamasi Diameter protokolo pagalba.

Atlikus skambučio įvykio patikrinimą ir prireikus modifikavimą, apdorojimas perduodamas kitoms operatoriaus programoms, funkcijoms.

Sistema veiks tik VoIP tinkle, bendraudama SIP protokolu. Ji turi veikti Mobicents JSLEE platformoje.

3.1.2. Vartotojai

Išskiriami du sistemos vartotojo lygiai:

- Sistemą diegiantis ir prižiūrintis asmuo.

Galima detaliau išskaidyti į du lygius: JSLEE platformos administratorius bei JSLEE paslaugų administratorius. Dažnai jų atliekamos operacijos persidengia, todėl paprastumo dėlei jų neskaidysime.

Vartotojo funkcijos. Šis asmuo, dažniausiai administratorius, prižiūri sistemos darbą, atlieka nesudėtingus konfigūracijos pakeitimus. Iškilus bėdai, jis turi surinkti pakankamai struktūrizuotos informacijos apie kilusią problemą, kad problemą sprendžiantys programuotojai (sistemos kūrėjas ar kita kontraktinė įmonė) galėtų ją sėkmingai lokalizuoti ir atlikti reikalingus pakeitimus.

Administratoriui turi būti lengvai prieinama sistemos konfigūracinė informacija.

Programos veiksmų registro detalumas privalo būti taip pat konfigūruojamas, o pats registras lengvai pasiekiamas. Šis žmogus visas savo funkcijas privalo atlikti be jokio programos kodo modifikavimo.

Patirtis. Šio lygio vartotojas yra gana prityręs, kadangi jam tenka gana atsakingas darbas – prižiūrėti ir instaliuoti sistemas, kurios tiesiogiai neša darbdaviui pelną. Jis dirba su panašiomis į kuriamą sistemomis/paslaugomis, tačiau neprivalo žinoti kiekvienos jų vingrybių. Šiam vartotojui turi būti sukurta išsami sistemos dokumentacija.

Papildomos vartotojo charakteristikos.

Konfigūracija bus pateikiama anglų kalba. Sistemos dokumentacija yra lietuviška. Trečių šalių komponentų dokumentacija yra išverčiama jų nuožiūra, bent angliška versija tikrai yra prieinama. Konfigūracija – angliška.

- Operatoriaus klientas - žmogus, kuris naudojasi kuriamą paslauga.

Funkcijos. Šis vartotojas su sistema bendraus mobilaus telefono pagalba. Paslauga inicijuojama surinkus numerį. Jei kažkuriuo paslaugos naudojimo metu iškils problema – klientas turi būti informuojamas.

Patirtis. Kad pasinaudoti paslauga, vartotojas turi turėti veikiantį SIP telefoną. Tam tinka tiek aparatinis, tiek programinis SIP telefonas. Naudojimas sistema apsiriboja reikalingos informacijos įvedimu. Iš vartotojo reikalaujama sugebėti įvesti reikiamus simbolius telefone – paskambinti norimu adresu. Tai gali būti ir asmuo pirmą kartą besinaudojantis telefono aparatu. Bet kuriuo atveju turi būti pateikiama detali dokumentacija, pagal kurią vartotojas gali iškviešti ir naudotis paslauga.

Papildomos vartotojo charakteristikos.

Kaip jau minėt anksčiau – vartotojas turi turėti veikiantį aparatinį ar programinį SIP telefoną. Be abejo, telefonas turi būti sukonfigūruotas ir prisijungti prie paslaugą teikiančio operatoriaus tinklo.

Paslaugos naudojimas neįgalių vartotojų priklauso nuo jų neįgalumo bei naudojamo SIP telefono galimybių. Paslaugos galutinis tikslas nesiskiria nuo paprasto skambučio.

Pati paslauga naujų pranešimų negražina. Naudojami operatoriaus, telefono pranešimai kai tikslas nepasiekiamas, yra užimtas... Paslauga turi užtikrinti jog atitinkami SIP pranešimai yra išsiunčiami, jų pranešimas ar ne priklauso nuo vartotojo naudojamo telefono,

jo konfigūracijos. Esant reikalavimui galima įgyvendinti klaidos pranešimą *TekstasĮKalbą* (*angl.* TextToSpeech) komponento pagalba.

Galimų vartotojų interesai neturėtų susikirsti, tačiau taip atsitikus *operatoriaus klientas* yra svarbesnis, kadangi jis neša įmonei pelną.

3.1.3. Įpareigojantys apribojimai

3.1.3.1. Apribojimai sprendimui

Paslaugos užklauso apdoravimo laikas (laiko tarpas tarp paslaugos gavimo ir atsakymo išsiuntimo užklausiui vartotojui) neturi viršyti 0,02 sekundės.

Paslaugos veikiamas numatytas tik domeno ribose.

Paslauga kuriama JSLEE platformoje (jNetX arba Mobicents).

Skambučiai vykdomi tik VoIP (SIP) tinkle

3.1.3.2. Diegimo aplinkai

Planuojama sistemą diegti atskirame Linux serveryje.

Sistemos konfigūracija neprivalo būti keičiama jos vykdymo metu.

Paslauga bendraus su kitais JSLEE komponentais, pačia JSLEE platforma, duomenų baze bei operatoriaus apmokėjimo sistema. Reikia užtikrinti, kad aplinkoje veiktų JSLEE platforma, visi komponentai būtų įdiegti, o reikalingi išoriniai komponentai (duomenų bazė bei apmokestinimas) – prieinami.

3.1.4. Svarbūs faktai ir prielaidos

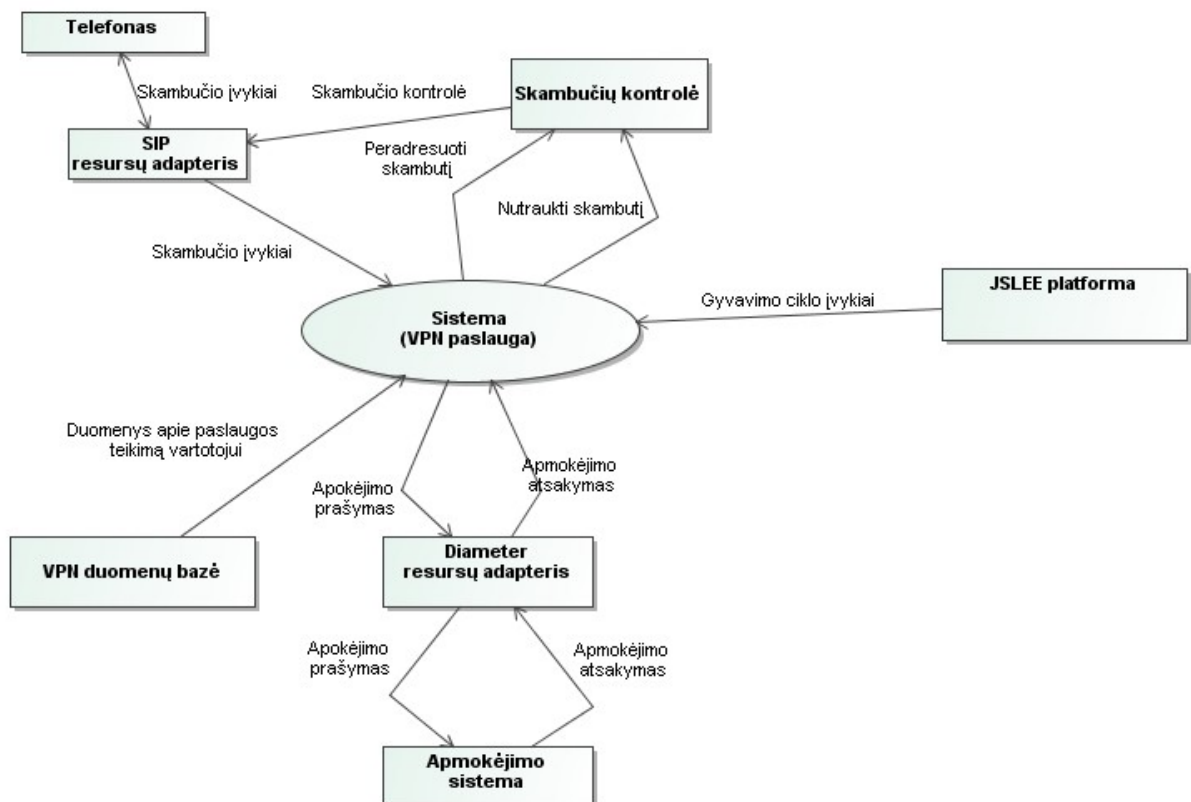
Prielaidos:

- Bendradarbiaujančios sistemos, į kurias kreipiamasi, atsako, atsako laiku ir teisingai (pagal sistemų specifikaciją)
- Paslaugos vartotojas – realus operatoriaus klientas su veikiančiu telefonu.
- Bendravimas tarp sistemų vyksta akimirksniu (rezultatas gaunamas greičiau nei per 0.02 sekundės)
- Apmokestinimo galimybės priklausys nuo reikalingų komponentų išbaigtumo (priklauso nuo Mobicents teikiamų komponentų). Šiuo metu, Diameter resursų adapterio dokumentacija nėra pilna, nėra išsamių pavyzdžių. Kas įtakoja apmokestinimo išbaigtumą.

- Apmokestinimas bus išbandomas su „Ericsson: Diameter Charging SDK 1.0“
- IMS registravimo komponentų naudojimas nėra būtinas. Tikslas yra išbandyti telekomunikacijų paslaugos kūrimą. Ji bus sukurta Mobicents platformai, kuri atitinka IMS aplikacijų serverį. Įtraukimas į pilną IMS architektūrą nereikalauja papildomo funkcionalumo pridėjimo. Reikia papildomų resursų konfigūravimui, instaliavimui... kas nesuteikia tai pateisinančios naudos.

3.1.5. Veiklos sfera

3.1.5.1. Veiklos kontekstas



4 pav. Konteksto diagrama

Bendradarbiaujančios sistemos jau buvo paminėtos ir šiek tiek aptartos 3.1.4. punkte. Beveik su visais komponentais paslauga bendrauja per JSLEE įvykių pagalbą. Kuriama programa susideda iš dviejų dalių SBB ir Resursų adapterio. SBB veikia tik JSLEE ribose. RA bendrauja tik su VPN duomenų baze ir perduoda užklaimus atgal į JSLEE.

3.1.5.2. Bendradarbiaujančios sistemos

Kaip jau minėta, paslauga kuriama JSLEE platformoje. Jos tikslas iškviešti komunikuoti su turimomis operatoriaus sistemomis skambučių, sąskaitos kontrolei. Šios bendradarbiaujančios, operatoriaus jau turimos sistemos yra iškviečiamos JSLEE pagalba.

Galima išskirti šias bendradarbiaujančias sistemas, komponentus:

- duomenų bazė – kuriamos paslaugos duomenų saugojimo ir gavimo vieta. Kuriant bus naudojama nauja, atskira, tačiau operatorius gali integruoti į savo.
- skambučio kontrolė – gaunami skambučio eigos įvykiai. Paslauga gali reaguoti į įvykius, jais įtakoti kitus komponentus, kurti naujus, kad pasiekti norimą tikslą.
- SIP resursų adapteris – atsakingas už signalų/įvykių perdavimą tarp SIP telefonų ir JSLEE.
- Diameter resursų adapteris – bendravimui su operatoriaus apmokestinimo komponentu.
- Apmokėjimo komponentas – operatoriaus sistema atliekanti apmokestinimą. Su ja JSLEE bendrauja Diameter protokolu.
- JSLEE platforma – joje veiks sukurta paslauga.
- VPN resursų adapteris – bendraus su naudojama duomenų baze.
- VPN SBB – kontroliuos paslaugos vykdymą – skambučių nukreipimą.

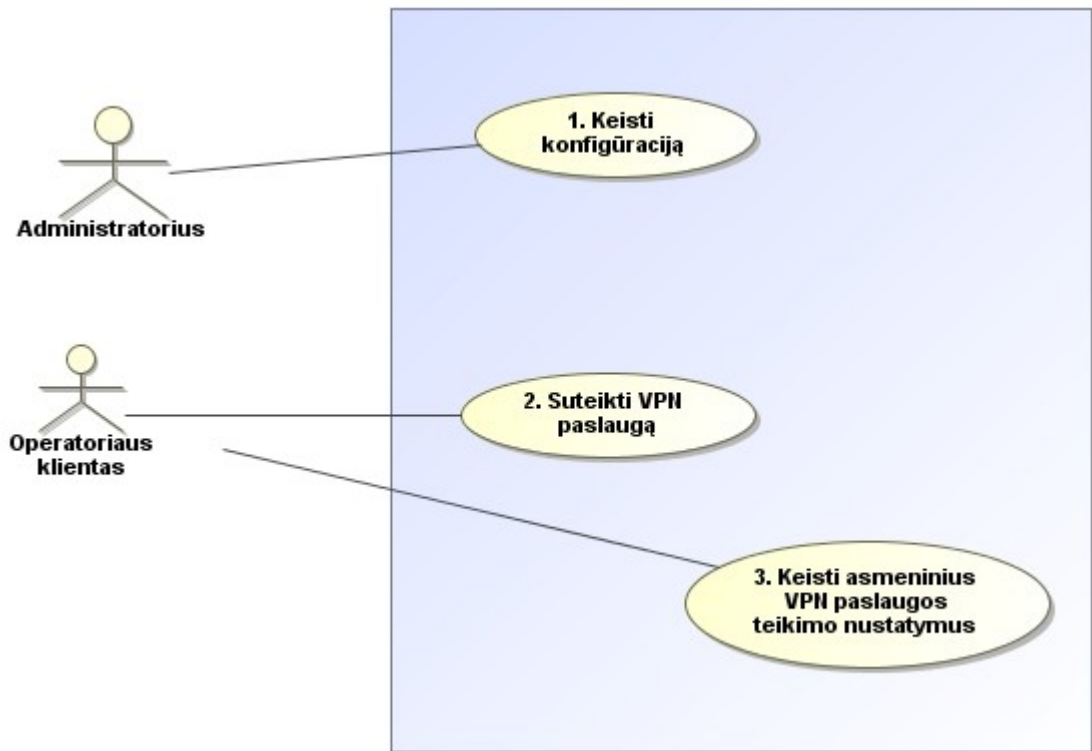
3.1.5.3. Veiklos padalinimas

3 lentelė: Veiklos įvykių sąrašas

Eil. Nr.	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1	Duomenys apie paslaugos teikimą vartotojui	Iš duomenų bazės paimti duomenys apie vartotoją ir jo VPN paslaugos konfigūraciją (į)
2	Apmokėjimas	Prašymas apmokėti paslaugą (iš) Apmokėjimo rezultatas (į)
3	Gyvavimo ciklas	Gyvavimo ciklo įvykiai (į)
4	Nutraukti skambutį	Nutraukti skambutį (iš)
5	Peradresuoti skambutį	Peradresuoti skambutį (iš)

3.1.6. Produkto veiklos sfera

3.1.6.1. Sistemos ribos



5 pav. Panaudojimo atvejų diagrama

Paslauga nėra sudėtinga. Joje dalyvauja du aktoriai (apibrėžti 3.1.2.) skyrelyje.

Galimi paslaugos panaudos atvejai:

1. *Sistemos konfigūracijos keitimas.* Administratorius keičia sistemos veikimo nustatymus: komunikuojančių sistemų adresus, paleidimo parametrus, registro nustatymus... Reikalavimuose apibrėžta, kas turi būti reguliuojama. Paslaugos konfigūravimas neturi reikalauti programuotojo įsikišimo ir vykti pagal JSLEE numatytas procedūras jei nenumatyta kitaip.
2. *Trumpų numerių paslaugos (VPN) suteikimas.* Paslauga vartotojui teikiama automatiškai, priklausomai nuo jos konfigūracijos vartotojo bei operatoriaus lygmenų.
3. *Asmeninių paslaugos nustatymų keitimas.* Vartotojas gali konfigūruoti paslaugos teikimą per operatoriaus pateikiamą prieigą. Sukuriamas demonstracinis internetinis puslapis, kuriame galima reguliuoti vartotojų paslaugos teikimo nustatymus: veikimą ir ne, trumpų numerių administravimas... (detaliau reikalavimus apibrėžiančiuose skyreliuose 3.1.7. ir 3.1.8.)

3.1.7. Funkciniai reikalavimai

1. Skambinantis asmuo nemato, kur nukreipiamas trumpas numeris
2. Paslauga nesuteikiama/neapmokestinama, jei skambinamas numeris nepasiekiamas. Jei numeris, kuriam vartotojas nori paskambinti yra nepasiekimas (ne ryšio zonoje, tokio numerio nėra ar telefonas išjungtas) – paslauga nesuteikiama/neapmokestinama.
3. Galima laikinai išjungti paslaugos teikimą. Vartotojas, jam suteiktos konfigūracinės vartotojo sąsajos pagalba, gali nustatyti ar jis naudojami VPN paslauga. Persigalvojus, galima atstatyti į buvusią padėtį.
4. Trumpojo numerio galiojimą galima laikinai sustabdyti. Sustabdžius vieno ar kelių trumpų numerių galiojimą. Kiti yra teikiami normaliai, pagal globalius paslaugos nustatymus.
5. Galimybė sukurti savo numeriui sutrumpintą. Paslaugos naudotojui galima prisiskambinti VPN trumpaisiais numeriais.
6. Galimybė susikurti daugiau nei vieną trumpąjį numerį. Vartotojas gali susikurti kelis trumpus numerius, kuriais jis yra pasiekiamas.
7. Operatorius gali riboti kiek trumpų numerių vartotojas gali susikurti.
8. Administratorius gali reguliuoti įvykių žurnalo pildymo detalumą.
9. Instaliavus kuriamą sistemą naujoje vietoje ar pakeitus vienos iš bendradarbiaujančių sistemų prisijungimo vartotoją, adresą, administratorius turi galėti lengvai pataisyti susijungimą tarp šių sistemų.
10. Paslauga teikiama tik tam pačiam SIP domene. Norint, kad trumpas numeris iš kito domeno veiktų, reikia kartu nurodyti ir tikslo domeną skambinant.
11. Paslaugos apmokestinimui programa Diameter protokolu kreipiasi į operatoriaus apmokėjimų sistemą.
12. Paslaugos apmokestinimo būdas yra nurodomas operatoriaus. Apmokestindama paslaugą programa atsižvelgia į vartotojo apmokėjimo nustatymus.

3.1.8. Nefunkciniai reikalavimai

- a) Reikalavimai sistemos išvaizdai
 - Sistema neturės jokios vartotojo sąsajos. Įvedami duomenys bei iš sistemos gautų žinučių rodymas priklausys nuo vartotojo naudojamos aparatūros ir jos nustatymų.
- b) Reikalavimai panaudojamumui

- Kuriama sistema negalės turėti didelės įtakos panaudojimo paprastumui. Daugumoje atvejų tai priklausys nuo operatoriaus ir naudojamo telefono.
 - Sukonfigūravus VPN paslauga ji teikiama automatiškai. Veikianti paslauga neįtakoja įprasto naudojimosi telefonu.
- c) Reikalavimai vykdymo charakteristikoms
- Laiko tarpas nuo vartotojo duomenų gavimo iki atsakymo nusiuntimo neturi viršyti 0,02 sekundės.
- d) Reikalavimai sistemos priežiūrai
- Gali keistis bendradarbiaujančių sistemų priėjimas: pasikeisti internetinio serviso tipas, sistema bendraus dvejetainiu kanalu... Kuriant sistemą į tai turi būti atsižvelgta.
 - Paslaugos konfigūracijos administravimas turi būti atliekamas pagal JSLEE specifikaciją.
- e) Reikalavimai saugumui
- Reikalavimų saugumui nėra. Duomenų bazės ir JSLEE saugumu turi pasirūpinti administratorius

3.1.9. Naujos problemos

- Problemos diegimo aplinkai

Kuriama sistema priklauso nuo kitų sistemų darbo, taigi turi būti užtikrinta jog visos bendradarbiaujančios sistemos veikia, o kuriama programa gali jas rasti, prisijungti ir gauti reikalingus duomenis.

- Įtaka jau instaliuotoms sistemoms

Kuriama paslauga įeina į operatoriaus naudojamų sistemų rinkinį. Ji naudojami komponentų teikiamais resursais: išskviečiamos kitos sistemos, gaunami iš jų duomenys, tačiau kitos įtakos joms nedaroma.

- Neigiamas vartotojų nusiteikimas

Vartotojų nepasitenkinimą gali sukelti operatoriaus naudojama kainodaros politika, trumpų numerių pasirinkimo skaičius, galiojimo ribojimas tik vidinio SIP domeno ribose.

- Galimos naujos sistemos sukeltos problemos

Kuriamai sistemai turi būti prieinamos visos bendradarbiaujančios sistemos bei reikalingos jų funkcijos. Tam pasiekti gali prireikti pakeisti tinklo struktūrą, sukurti ar padaryti naujus priėjimus, gal netgi modifikuoti jau veikiančias sistemas.

3.1.10. Rizikos

3.1.10.1. Galimos sistemos kūrimo rizikos

4 lentelė: Galimos rizikos ir jų įvertinimas

Rizika	Tikimybė
Per ilgai trunkantys skaičiavimai	0,1
Blogai sudarytas darbų tvarkaraštis	0,05
Neveikiančios bendradarbiaujančios sistemos	0,25
Klaidingai surinkti reikalavimai	0,2
Neteisingai užrašyti reikalavimai	0,15

3.1.10.2. Atsitiktinumų valdymo planas

Jei problema iškyla su kodu – ją taisy programuotojas. Tam skiriama papildomai laiko dokumentacijos sąskaita.

Jei neveikia bendradarbiaujančios sistemos – surenkami reikalingi duomenys ir pateikiamas prašymas už jas atsakingus žmones atlikti reikalingus pakeitimus.

Radus klaidą darbų tvarkaraštyje bandoma ją pataisyti apie taisytinus variantus pranešant projekte dalyvaujantiems asmenims.

Atradus problemą tarp reikalavimų ji aiškinamasis su užsakovu. Kol tai vyksta kiti darbai su probleminiu reikalavimu neatliekami.

3.1.11. Vartotojo dokumentacija ir apmokymas

Bus pateikiamos dvi dokumentacijos:

- *Sistemos konfigūravimo* – administratoriui skirta dokumentacija. Joje aprašoma kaip instaliuoti bei konfigūruoti sistemą. Vartotojas galės reikšti pageidavimus ir neaiškumus į kuriuos bus atsižvelgta. Tai žemo lygio dokumentacija: vartotojas turės susigaudyti kaip diegiamos ir prižiūrimos panašios sistemos. Dokumentacijos pateikimo forma – pdf dokumentas.
- *Paslaugos naudojimo* – skirta operatoriaus vartotojui. Joje aprašoma kaip naudotis paslauga. Vartotojas dokumentacijos rašyme nedalyvaus. Tai aukšto lygio dokumentacija: visi žingsniai paaiškinami išsamiai, detalai ir su vaizdiniais

pavyzdžiais. Dokumentacijos pateikimo forma – *pdf* dokumentas. Operatorius gali vėliau ją modifikuoti, spausdinti, perkelti į internetinį puslapį ar kitaip manipuluoti.

3.1.12. Perspektyviniai reikalavimai

- Sistemos naudojamų resursų įvertinimas
- Pilnas IMS architektūros palaikymas
- GSM ir kitų telekomunikacijų tinklų palaikymas
- VPN konfigūracijos portalo sukūrimas
- Duomenų bazės integracija su HSS
- Ro/Rf apmokestinimas
- Trumpų numerių naudojimo statistinis registravimas ir pranešimas vartotojui
- Migravimas į JSLEE 1.1 specifikaciją atitinkančią platformą

3.2. Architektūros specifikacija

3.2.1. Architektūros pateikimas

Kuriamos sistemos architektūra yra pateikiama pagal KTU „Programų sistemų inžinerijos magistrantūros“ rekomenduojamas gaires. UML sistemos aprašymas išskiriamas į sekančius penkis vaizdus:

- *Panaudos atvejų vaizdas*. Apibūdina veikėjus ir sistemos panaudos atvejus. Šis vaizdas pateikia vartotojo poreikius, o žemesniuose modelio lygiuose atvaizduoja atskirų srautus ir apribojimus.
- *Sistemos statinis vaizdas*. Sistemos sritis išskaidoma į atskirus objektus, kurie vėliau sugrupuojami pagal tam tikrus panašumus .
- *Sistemos dinaminis vaizdas*. Apibūdinama kaip kinta sistemos būsenos jos vykdymo eigoje, kokie komponentai bendrauja, kokia informacija perduodama ir pan. Šiame vaizde bandoma parodyti kokia veiksmų seka turi vykti kiekvieno panaudos atvejo metu,
- *Išdėstymo vaizdas*.

Kiekvienas šių vaizdų yra susijęs ir paveldi bei praplečia prieš tai buvusio vaizdo funkcionalumą.

3.2.2. Architektūros tikslai ir apribojimai

Užsakovas daugumoje savo produktų naudoja JSLEE platformą. Iki šiol pasirinkimas buvo jNetX gaminys, tačiau vis daugiau klientų, o kartu ir pats užsakovas, išreiškia susidomėjimą atviro kodo JSLEE specifikaciją atitinkančiu produktu Mobicents. Pranašumų ar skirtumų tarp šių platformų nenagrinėsime šiame dokumente. Iš architektūrinio požiūrio esminių skirtumų tarp šių platformų nėra, kadangi abi atitinka JSLEE 1.0 specifikaciją. Naujesnė Mobicents versija netgi atitinka 1.1 specifikacijos versiją.

Kitas svarbus apribojimas – realizuojama tik VoIP, SIP paremta paslauga.

Apmokėjimui naudojamas atskira sistema su kuria bendraujama Diameter protokolu.

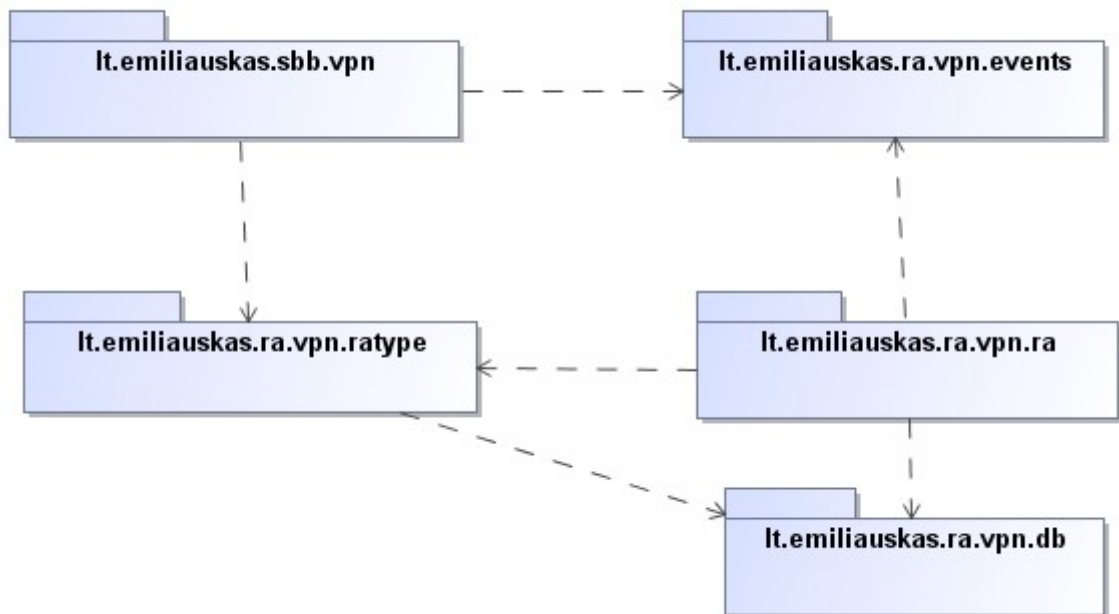
Paslaugai nėra keliamas reikalavimas, kad ji veiktų pilnoje IMS architektūroje. Kol tai rekomenduotina praktika, tačiau konfigūravimas gali pareikalauti daug pastangų, o jos suteikiamais privalumais, šiame paslaugos vystymo etape nepasinaudosime.

Sistema neprivalo veikti keliuose kompiuteriuose. Pritrūkus resursų ji būtų įdiegiama keliuose kompiuteriuose, o prieš juos pastatomas krūvio skirstytojas. Tačiau yra numatyta, jog prie tam tikrų sąlygų bendradarbiaujanti sistema galėtų nutraukti kuriamos paslaugos tiekimą.

Darbo grupę sudaro vienas žmogus – autorius. Užsakovas ir vadovas yra konsultaciniai asmenys.

3.3. Sistemos statinis vaizdas

3.3.1. Apžvalga



6 pav. Sistemos paketai

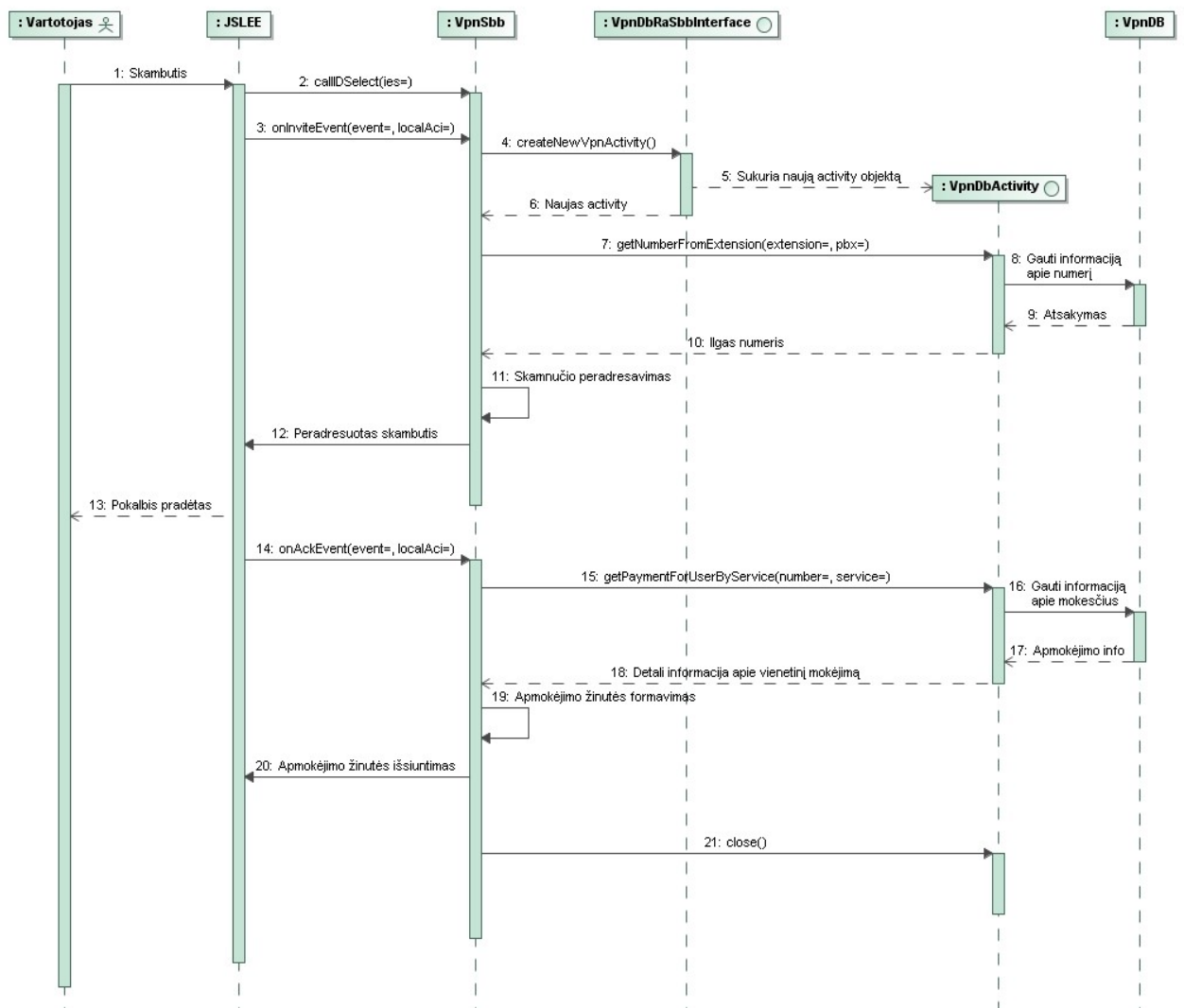
Punktyrinė linija parodo ryšį tarp paketų. Rodyklė parodo kurio paketo klasė naudoja kito paketo klases. Detalesnė informacija apie paketus yra pateikta sekančiame skyrelyje 3.3.2. Paketų detalizavimas.

3.3.2. Paketų detalizavimas

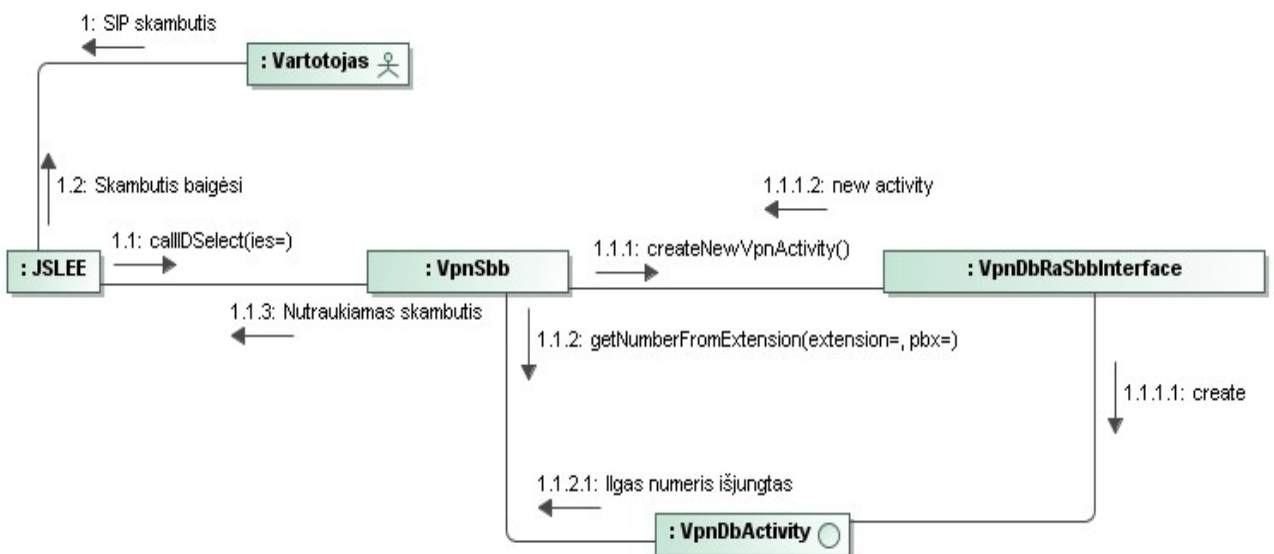
Pateiksime trumpus sistemos paketų aprašymus bei į juos įeinančias klases.

- *lt.emiliauskas.sbb.vpn* – čia kuriamos SBB (Service Building Block) klasės. Šiame pakete esančios klasės atsakingos už paslaugos davimą: skambučio stebėjimą, peradresavimą, duomenų iš kitų komponentų gavimą.
- *lt.emiliauskas.ra.vpn.events* – čia talpinami VPN resursų adapterio įvykiai.
- *lt.emiliauskas.ra.vpn.ratype* – tai VPN resursų adapterio API paketas. Šio paketo turėtų pakakti bet kuriam komponentui norint bendrauti su VPN RA. Pagrindiniai komponentai – interfeisai, tačiau gali būti ir klasių, kurioms interfeisų kurti neapsimoka (įvykiai).
- *lt.emiliauskas.ra.vpn.ra* – VPN resursų adapterio ir jo pagalbinės klasės bei kiti komponentai.
- *lt.emiliauskas.ra.vpn.db* – prisijungimo prie duomenų bazės logika.

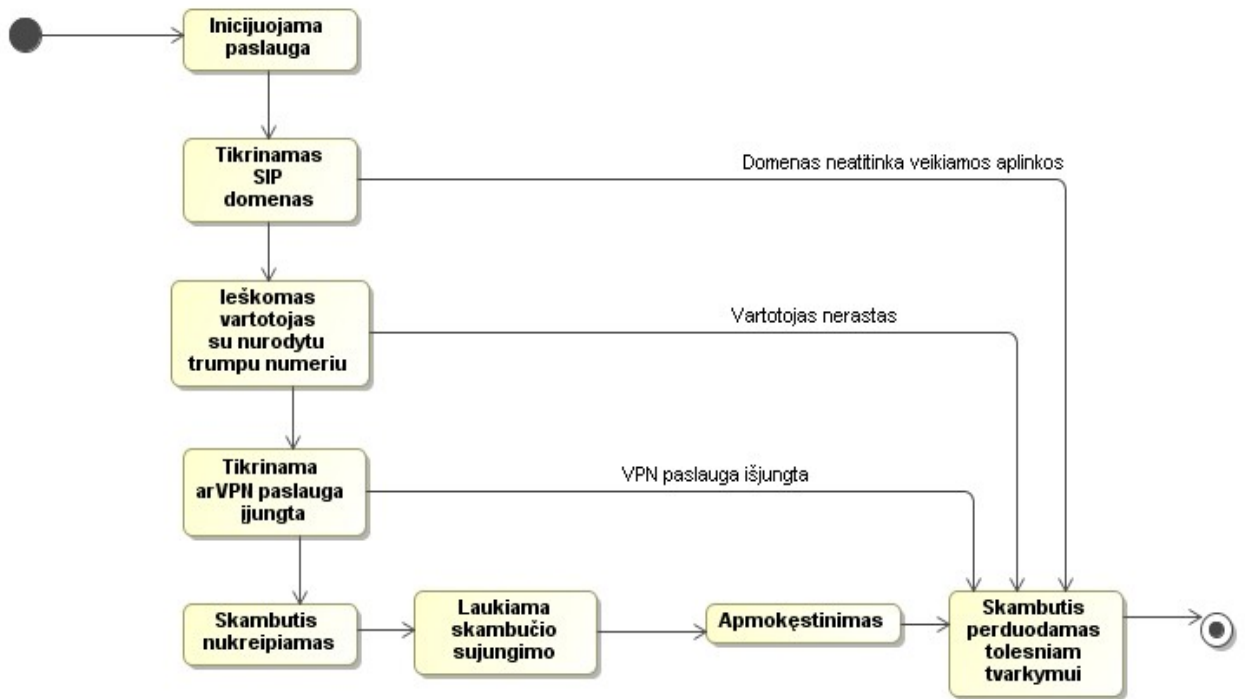
3.4. Sistemos dinaminis vaizdas



7 pav. Sekų diagrama kai paslauga suteikiama be nesklandumų



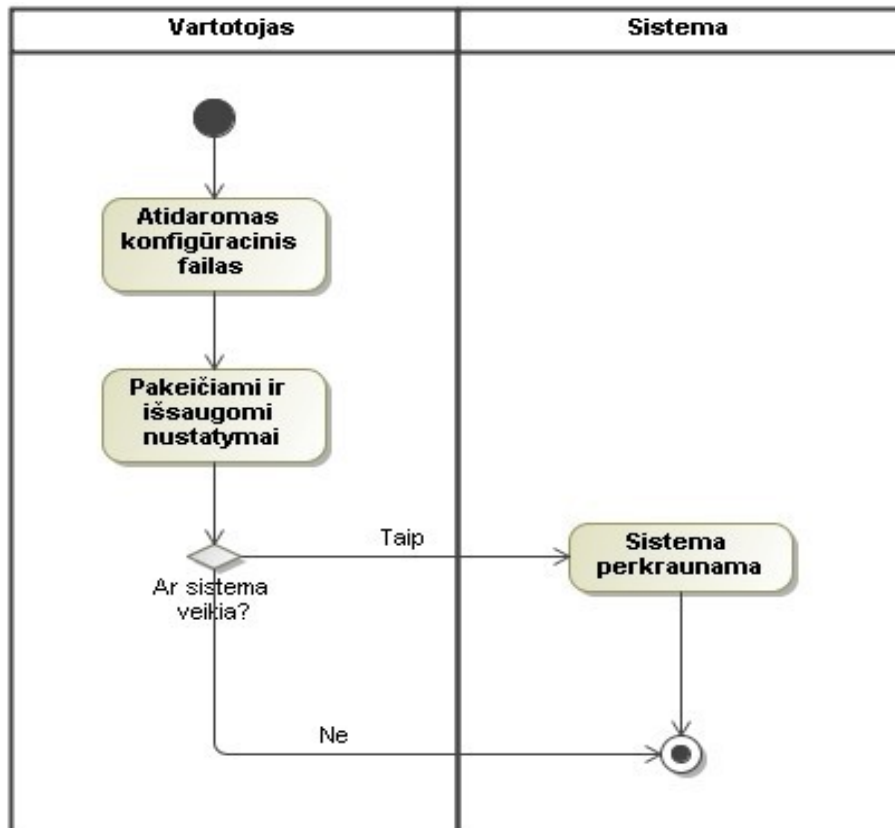
8 pav. Bendradarbiavimo diagrama, kai paslauga nesuteikiama dėl negaliojančio trumpo numerio



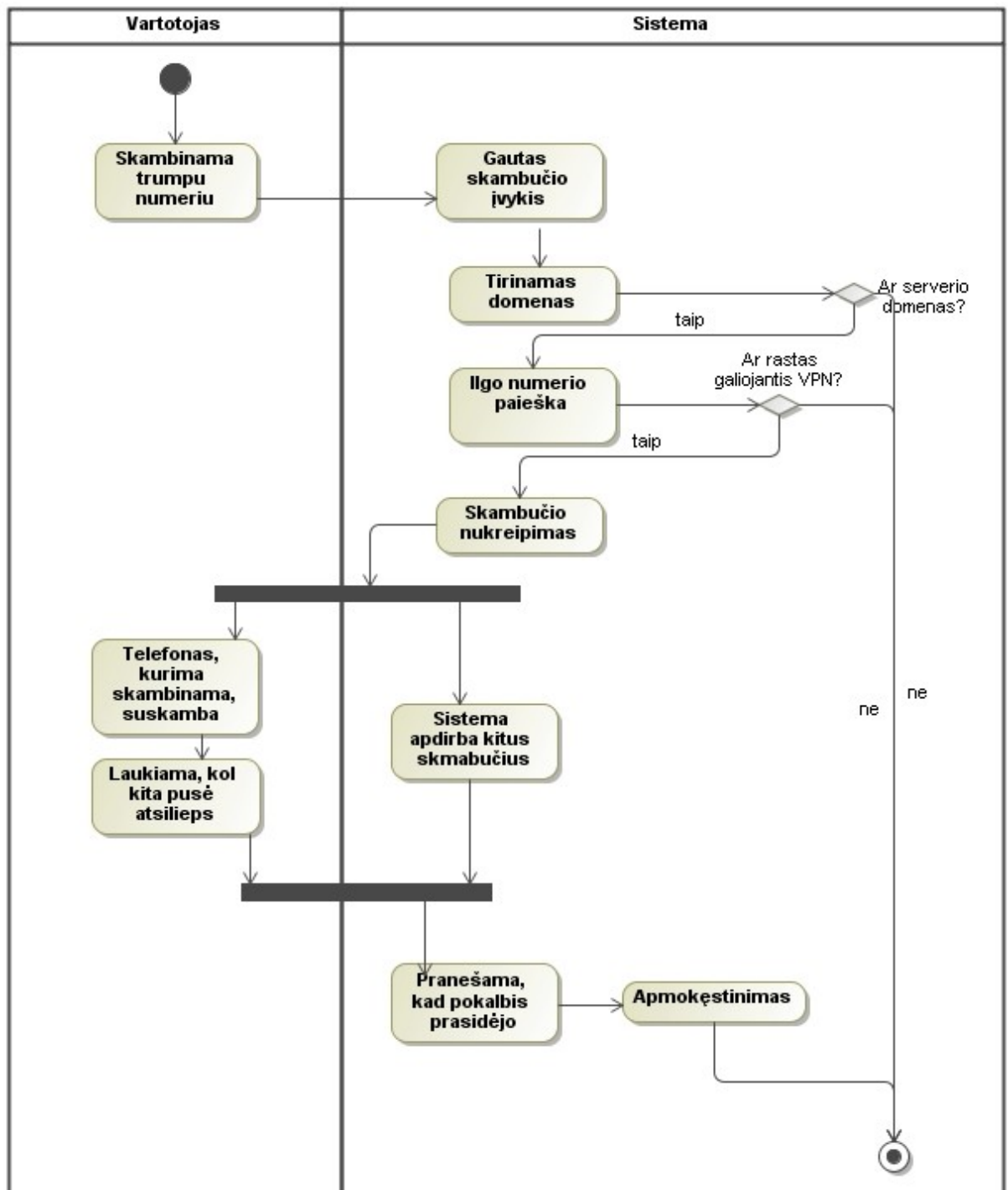
9 pav. Paslaugos teikimo būsenų diagrama



10 pav. Sistemos konfigūracijos valdymo būsenų diagrama

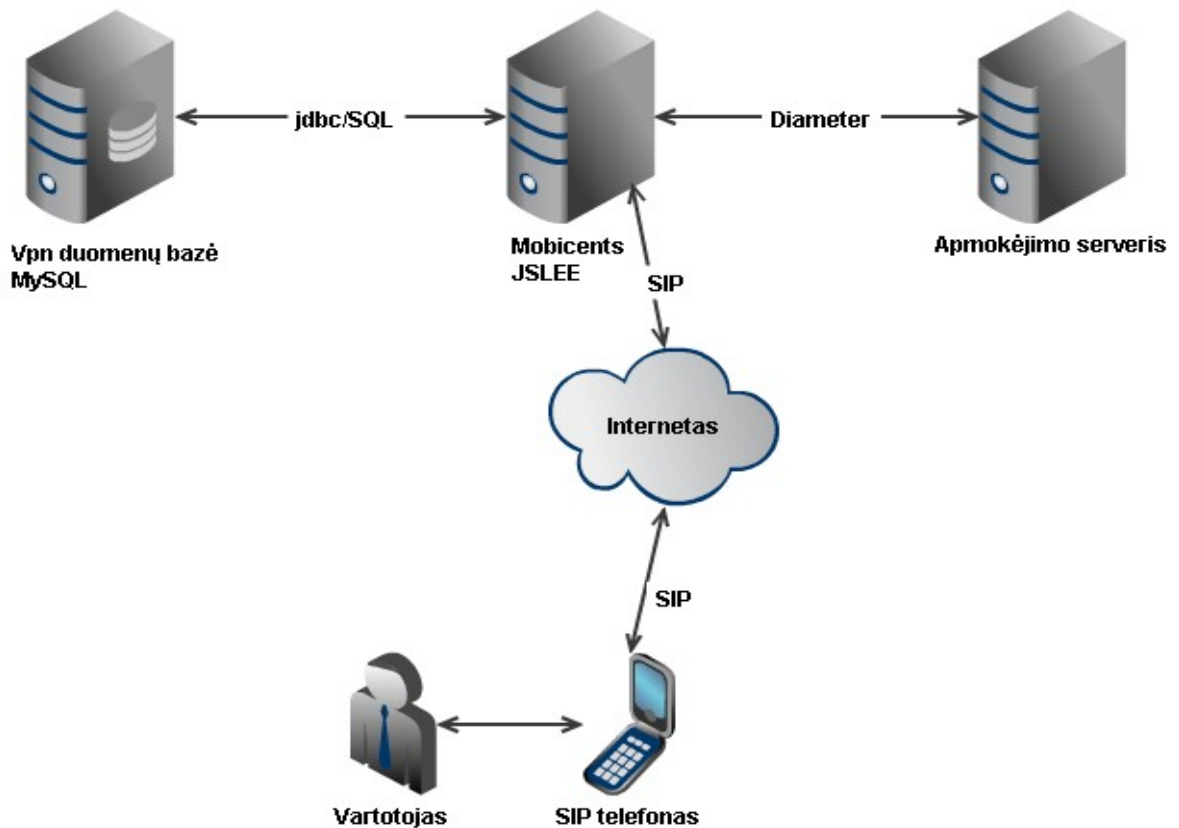


11 pav. Sistemos administravimo veiklos diagrama



12 pav. Paslaugos teikimo veiklos diagrama

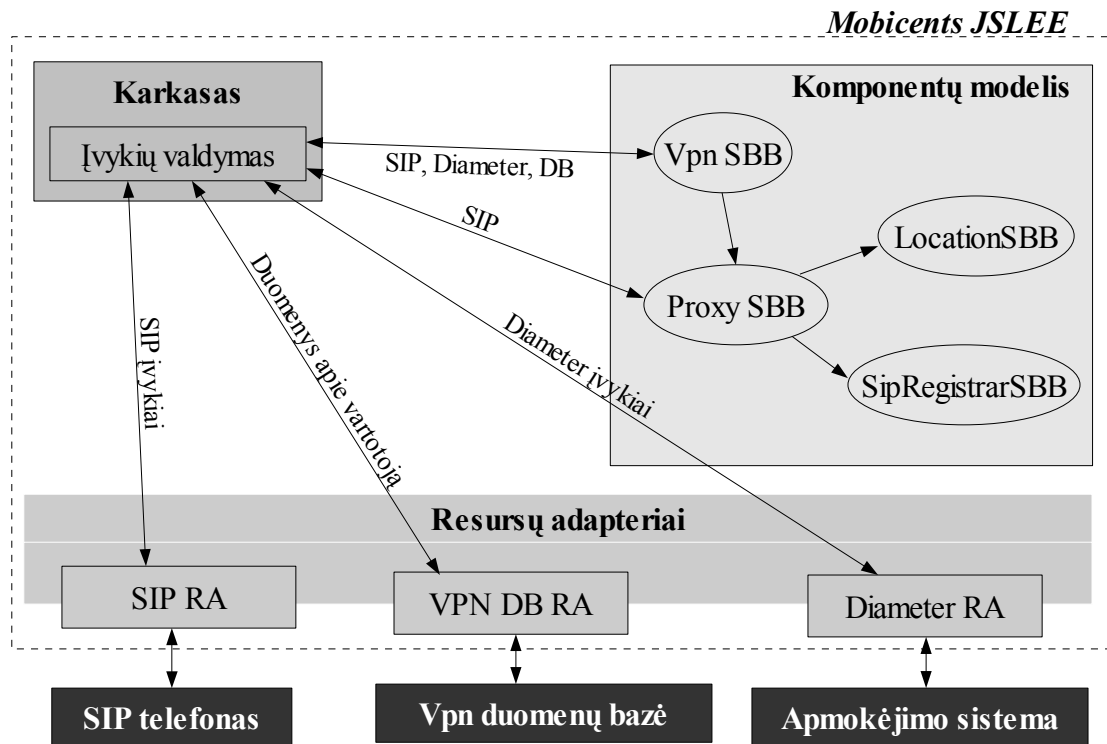
3.5. Išdėstymo vaizdas



13 pav. Išdėstymo vaizdas

Išdėstymo vaizdo komponentai:

- SIP telefonas – telefonas, kuriuo vartotojas kreipiasi į sistemą. Tai gali būti aparatinis ar programinis, svarbiausiai, kad pilnai palaikytų SIP protokolą. Kuriamos paslaugos komponentas su juo bendrauja per JSLEE veikiančią SIP resursų adapterį.
- JSLEE serveris – serveris, kuriame veikia kuriama paslauga (Mobicents platforma). Čia veikia abu realizuojami komponentai: resursų adapteris ir SBB.
- VPN duomenų bazė – čia saugomi paslaugai reikalingi duomenys, vartotojo konfigūracija. Ji nėra didelė ir gali būti integruota į kitą tinkle naudojamą duomenų bazę (HSS, apibendrinta JSLEE duomenų bazė ir pan.). Į ją kreipiasi sukuriamas resursų adapteris.
- Apmokėjimo serveris – kontroliuoja vartotojo sąskaitą, leidžia apmokestinti teikiamas paslaugas, gauti sąskaitos balansą... Į čia per Diameter resursų adapterį kreipiasi realizuotas SBB.



14 pav. JSLEE specifikacijos architektūra

14 pav. pavaizduota vidinė Mobicents JSLEE aplinka kurioje veikia paslauga. Kaip jau minėta produktas bus sudarytas iš dviejų programinių komponentų: VpnSbb ir VpnDb resursų adapterio.

JSLEE platformoje SBB komponentai (diagramoje ovalai) atsakingi už paslaugos teikimą – SLEE generuojamų įvykių apdorojimą ir kartais generavimą.

Įvykiai diagramoje atvaizduoti linijomis su rodyklėmis. Rodyklės parodo ar įvykiai gaunami, ar sukeliami, ar abu atvejai. Šiuo atveju visiems naudojamiems komponentams tinka paskutinis atvejis (įvykiai gaunami ir sukuriami). Ant linijos parašytas tekstas nurodo kokia informacija/įvykiai ja keliauja.

Resursų adapteriai atsakingi už bendravimą su išore. Jiems priklauso visi įvykiai ir yra pagrindinei jų generatoriai: interpretuojant aplinkos būvį apie pokyčius pranešama JSLEE įvykių pagalba. Taip pat galimas atvirkštinis variantas: JSLEE komponentai praneša apie resurso būvį arba prašo išorinio komponento duomenų, duoda komandą... Tokiu atveju per atitinkamą resursų adapterį apie tai pranešama išoriniam resursui JSLEE įvykių pagalba.

Išoriniai resursai dirba nepriklausomai nuo JSLEE. Visa informacija reikalinga resursų adapteriams susisiekti su jais yra saugoma konfigūracijoje JSLEE aplinkoje.

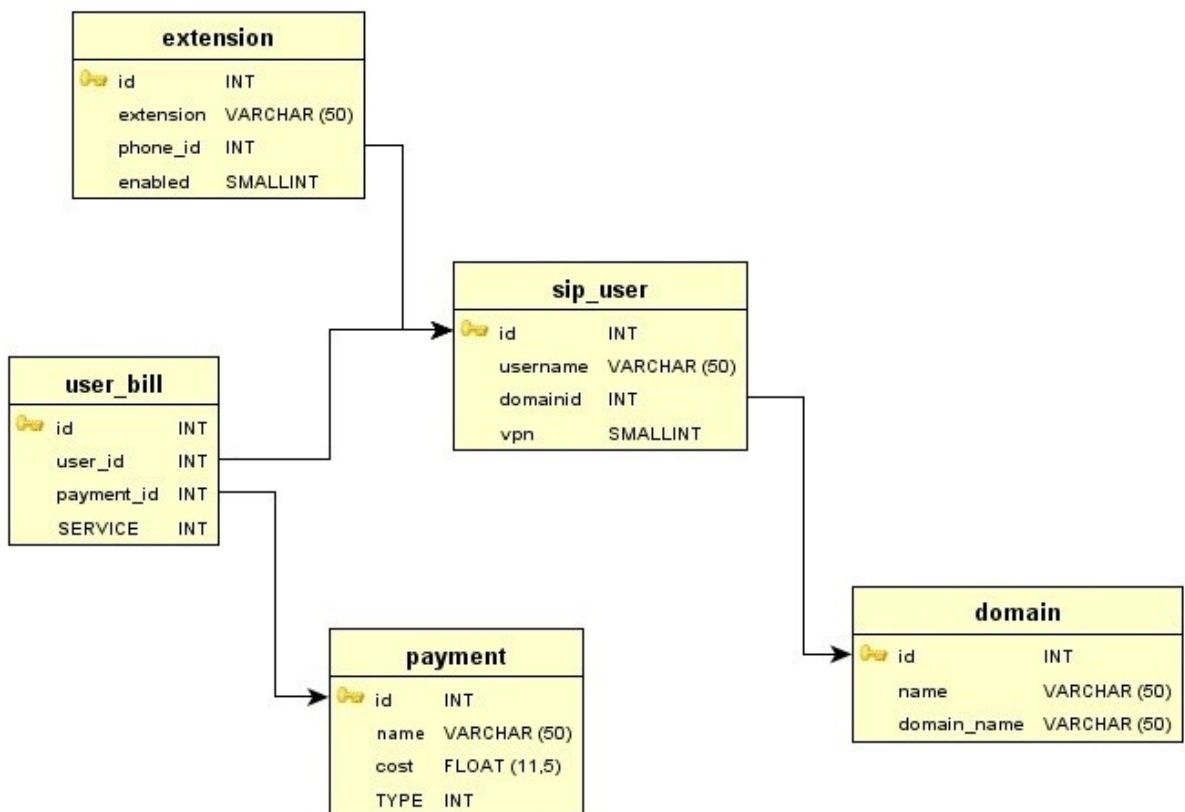
3.6. Duomenų vaizdas

Konfigūraciniai failai saugomi tekstiniuose failuose.

JAVA klasės yra sukompilijuojamos ir talpinamos *jar* failuose. Dauguma pagal specifikaciją yra talpinami į kitus *jar* ir t.t. Šios operacijos atliekamos pagal JSLEE specifikacijoje apibrėžtas taisykles su reikalingais papildomais failais.

Duomenų bazės gamintojas gali būti pasirenkamas. Kūrimui naudojama MySQL. Jai ir bus pateikiami sukūrimo skriptai.

Ši duomenų bazė yra naudojama tik paslaugos, operatoriui rekomenduojama susikurti atitinkamas lenteles į savo naudojamą schemą integruojant su turimomis apmokėjimo bei vartotojų (HSS) duomenų bazėmis.



15 pav. Sistemos naudojamos duomenų bazės diagrama

3.7. Kokybė

JSLEE yra didelio našumo bei žemo gaisties laiko įvykių apdorojimo aplinka. Kadangi telekomunikacijose daug reiškia teikiamų paslaugų atlikimo sparta, ši aplinka pateisina keliamus reikalavimus. Nors JSLEE nebūtina skirta tik telekomunikacijų sričiai, tačiau specifikaciją atitinkančios platformos yra pritaikytos būtent jai.

Viena iš alternatyvių platformų – atviro kodo Mobicents, yra vis dar kuriama tobulinama ir jos panaudojimas komerciniams tikslams gana ribotas: dauguma reikalingų komponentų yra nesukurti ar pilnai neištestuoti, nedokumentuoti. Tačiau jei ketinama apsiriboti mažesne sritimi (kuriamos paslaugos atveju) – platformos patikimumas ir kokybė yra pakankama ir mažai nenusileidžia komerciniams atitikmenims.

Projektuojama, kad paslauga bus prieinama dideliame vartotojų skaičiui. Pradžiai paslaugai keliami standartiniai telekomunikacijų spartos reikalavimai – operacijos atlikimo laikas neturi viršyti 0,02 sekundės.

Kadangi tiek platforma, tiek sistema parašyta Java programavimo kalba – jos paleidimas skirtingose operacinėse sistemose nėra sudėtingas. Serveryje veiks Linux operacinė sistema, o paslauga bus kuriama Windows arba Linux operacinėje sistemoje. Šiuo atveju migravimas tarp skirtingų aplinkų bus nesudėtingas.

4. PASLAUGOS TOBULINIMO GALIMYBĖS

Realizuota VPN telekomunikacijų paslauga ištestuota ir pristatyta užsakovui. Pakol kas ji nėra naudojama ar pardavinėjama. Užsakovas vis dar tiria Mobicents platformą, bando ją įvertinti. Nepaisant to, nuo magistro darbo rašymo pradžios pasikeitė keletas aplinkybių, kurias įvertinus, galima pagerinti paslaugos kokybę ir jos charakteristikas.

Pirmiausiai reikėtų apibrėžti mus dominančius kriterijus:

- *Sparta.* Kaip jau minėta ne kartą ir užfiksuota nefunkciniuose reikalavimuose – pageidaujama, kad vartotojui atsakymas atkeliautų ne vėliau nei per 0,02 sekundės. Sistema veikia realiu laiku, dirba su garsu ir vartotojas gali pastebėti bet kokius trikdžius. Jei vartotojas nepatenkintas - paslaugos teikėjas praranda pinigus.

Laiko tarpas nuo tada kai vartotojas iššaukė įvykį (pvz. surinko numerį) iki tol kol gavo atsakymą (pvz. garsinis signalas, jog skambinama kitai pusei) užtrunka ilgiau. Tačiau į jį įeina ir kelionė tinklu ir visų paslaugų apdorojimo trukmė.

Mes užsibrėžiame, kad paslauga negali trukti ilgiau nei 0,02 sekundės. Jeigu atlikus modifikacijas, pakeitus naudojamus komponentus šis laikas sumažėja – tai vertiname teigiamai.

- *Per sekundę apdorojamų skambučių kiekis.* Šis skaičius yra labiau charakteringas platformai ir yra susijęs su jau aptartu spartos kriterijumi. Jo pokyčio dėka galime nustatyti kaip paslauga įtakoja platformos veikimą. Be to šio kriterijaus nauda yra aiškiau įsivaizduojama nei spartos pokytis.

Kaip jau yra minėta, reakcijos į vartotojo veiksmus greitis yra svarbus. Jį įtakoja tinklas, platforma, aplinka... Galime mažinti paslaugos skaičiavimo laiką kitų resursų sąskaita visai apie tai nežinodami. Tuo tarpu įvertinę per sekundę apdorojamų skambučių kiekį, galime palyginti skirtumą tarp matavimo kai teikiama paslauga su matavimu kai ji neteikiama.

Stengiamės, kad šis skaičius būtų kuo didesnis, drastiškai nepakeičiant kitų kriterijų ir neperžengiant jiems užsibrėžtų tikslų.

- *Naudojami aparatiniai resursai.* Čia svarbūs du faktoriai.

Pirmiausia, kol aukštas resursų naudojimas nėra blogai. Problemos atsiranda, kai tais pačiais resursais nori pasinaudoti kiti aplinkoje esantys veikėjai: duomenų bazė, JVM, operacinė sistema... Jie sumažina paslaugai ir jos platformai prieinamų resursų kiekį taip sumažindami jos kokybę.

Žvelgiant iš kitos pusės, dažnai kuriant programas galima pasirinkti sprendimo būdą realizuoti padidinant vieno ar kito resurso sąnaudas taip gaunant kitas teigiamas savybes, pagerėjimus. Pavyzdžiui duomenis saugoti duomenų bazėje, faile ar atmintyje. Priklausomai nuo naudojamos aparatinės įrangos, jos sąnaudų pasiskirstymas gali skirtis.

Numatome, kad ir didžiausio apkrovimo metu, procesoriaus ir atminties sąnaudų vidurkis neviršytų 85%. Vertiname vidutines reikšmes, o ne labai trumpus pokyčius resursų sunaudojimui.

- *Platformos pasirengimo įvertinimas.* Realizavus paslaugą jau yra suformuotas bendras įspūdis apie „Mobicents“ bendruomenę, dokumentaciją, įrankius. Nereikėtų pamiršti jog tai telekomunikacijų platforma, kuriai yra suformuoti bendri reikalavimai.

Priklausomai nuo galimybių reikia įvertinti ar „Mobicents“ yra pakankamai išvystyta, kad ji galėtų veikti realioje aplinkoje, programuotojai gali be didesnių keblumų realizuoti paslaugas, o vartotojai jomis naudotis.

Nuo magistro studijų pražios praėjo 1,5 metai. Iš pradžių, autoriaus žinios apie telekomunikacijų sritį buvo labai menkos. Taip pat pirmą kartą susidurta su JSLEE specifikacija. Pasirinkta „Mobicents“ platforma tai pat nesnaudė: išleista ne viena nauja versija su įvairiais patobulinimais.

Sekantys faktoriai gali įtakoti mus dominančius kriterijus:

- Nauja JSLEE specifikacijos versija
- Nauja „Mobicents“ platformos versija
- Duomenų kiekio įtaka. Jų saugojimo vietos ir formato pakeitimas

4.1. Naujos JSLEE versijos įvertinimas

Prieš pradėdant paslaugos realizaciją, 2008 liepos 14 dieną, buvo pabaigta JSLEE 1.1 specifikacija [17]. „Mobicents“ versija, sertifikuota šiai versijai, po kelių vėlavimų išėjo tik 2010 metais, kai paslauga buvo sukurta buvo pabaigta. Naujos „Mobicents“ versijos patobulinius ir naujoves aptarsime vėliau. Šiame skyrelyje išskirsime JSLEE 1.1 atneštas naujoves [22]:

- Resursų adapterių (RA) patobulinimai:
 - Resursų adapterių architektūra apibrėžia sąsają tarp JSLEE ir RA. Tai reiškia, jog standartizuota kaip RA pateikia API prie jo valdomo išorinio resurso. Ko pasekoje

RA ir paslauga gali būti perkelta į kitus JSLEE 1.1 versijos aplikacijų serverius ir sėkmingai juose veikti.

- RA gali naudotis pagalbinėmis SLEE priemonėmis (būsenos valdymas, žadintuvas...)
- RA gali įgyvendinti daugiau nei vieną RA tipą
- Kiti 1.0 patobulinimai:
 - įrašomi profiliai
 - nauji profilių indeksavimo būdai (telefono numeris, dalinis atitikimas, baziniai užklausų simboliai, ilgiausio priešdėlio atitikimas...)
 - daugiau lankstumo
 - įgyvendintas prašymas paslaugų sąveikavimui tranzakcijų modelyje

Iš visų paminėtų, mums svarbesni yra RA ir profilių patobulinimai. Apie profilius plačiau užsiminsim kalbėdami apie informacijos saugojimą.

Resursų adapterių pokyčiai leidžia sukurtiems paslaugos komponentams laisvai migruoti tarp JSLEE 1.1 platformų. Anksčiau jie nebuvo apibrėžti specifikacijoje ir įgyvendinimas priklausė nuo platformos kūrėjų. Pagal dabartinę padėtį, 1.1 palaiko „Mobicents“ ir „Rhino“. Užsakovo požiūriu tai labai didelės naudos neteikia, nes jis dirba su „jNetX“ (kuri vis dar nepalaiko 1.1 JSLEE versijos) ir „Mobicents“. Taigi migruoti nėra tarp ko. Tačiau, žvelgiant į ateitį, perėjimas į 1.1 leis neprisirišti prie vieno JSLEE platformos gamintojo. Kas palengvina platformos keitimą pasikeitus aplinkybėms ar reikalavimams.

Iš užsibrėžtų tiriamų kriterijų tai įtakoja tik „Mobicents“ platformos pasirengimą. Šis punktas jai nesuteikia daug pranašumų – dar labiau įsitikiname šios platformos atvirumu, juo labiau, kad komercinis „jNetX“ produktas netgi nėra paskelbęs jokių planų apie naujos versijos išleidimą. Tačiau pastebėti ir trūkumai. „Mobicents“ kūrėjai žadėjo naują versiją nuo 2009 vasaros, tačiau sugebėjo ją išleisti tik praėjus daugiau nei 1,5 metų nuo specifikacijos užbaigimo.

4.2.Naujos „Mobicents“ versijos įvertinimas

Magistro darbo kūrimo eigoje, buvo išleistos kelios JSLEE 1.0 „Mobicents“ versijos. Didelių pakeitimų tarp šių platformų nebuvo, todėl migracija buvo lengva, o galutinis produktas pristatytas su naujausia versija (1.x.y yra JSLEE 1.0 „Mobicents“ versijos formatas). Prieš išleidžiant JSLEE 1.1 „Mobicents“ versiją (2.x.y yra JSLEE 1.1 „Mobicents“

versijos formatas, tiksliau dabar yra 2.0.0 versija), senos tobulinimas buvo sustabdytas. Tai nereiškia jog jos kodas ištrinamas ir tampa nepasiekiamas. Tačiau tai reiškia, jog galime daryti prielaidas, kad „Mobicents“ pakol kas nėra naudojama pelnui gauti. Priimta, kad sena produkto versija yra dar kurį laiką palaikoma po naujos išleidimo. O šiuo atveju, kūrėjai nusprendė, kad toks palaikymas nėra reikalingas ir niekam nepakenks.

Su 2.0.0 „Mobicents“ versija atkeliavo kelios naujovės, svarbūs patobulinimai:

- „Mobicents“ tapo sertifikuota JSLEE 1.1 platforma
- Pagerėjo našumas [20]

Apie JSLEE 1.1 teikiamus privalumus jau aptarėme prieš tai buvusiame skyrelyje. Šiuo atveju „Mobicents“ yra sertifikuota specifikacijos platforma, kas reiškia, kad visi patobulinimai įgyvendinti.

Mums svarbiausiais pasikeitimas yra spartos pagerinimas. Remiantis jų matavimais [20], pagerėjimas yra nuo 175 iki 750 skambučių per sekundę. Vienintelis trūkumas yra tai, kad matavimai yra atlikti su skirtingo galingumo kompiuteriais ir skirtingais konfigūraciniais parametrais. Norėdami patikrinti šiuos duomenis, turėsime patys atlikti eksperimentus. Tuo pačiu bus galima įvertinti ir sukurtos paslaugos įtaką platformos našumui.

JSLEE specifikacijoje teigiama, kad 1.1 palaiko 1.0 komponentus. Tačiau kadangi resursų adapterių specifikacija pasikeitė, jų atgalinis palaikymas yra nutraukiamas [17, puslapis 2]:

„1.1 specifikaciją atitinkanti platforma turi tenkinti šiuos atgalinius reikalavimus:

...

- 1.1 specifikacijos resursų adapterio kontraktas pakeičia 1.0 kontraktą. JSLEE 1.1 specifikaciją atitinkantis resursų adapteris privalo įgyvendinti 1.1 resursų adapterio kontraktą ir neturi įgyvendinti 1.0 kontrakto“

Čia kontraktas – interfeisas, kurį privalo įgyvendinti resursų adapterio klasė. Taigi, atitinkama versija palaikys tik jos resursų adapterio interfeisą.

Specifikacija mini, jog SBB modifikacija turi būti minimali (mūsų realizuotos paslaugos atveju jų neturėtų būti). Pagrindiniai pakeitimai susiję su konfigūracijos keitimu. Taigi norint, kad paslauga veiktų 1.1 platformoje, minimaliai reikės modifikuoti jos resursų adapterį bei atnaujinti konfigūracinius failus.

„Mobicents“ dokumentacijoje apie atgalinį palaikymą pakol kas nėra aprašyta. Remiantis, tuo kad specifikacija yra apibrėžus 1.1 => 1.0 atgalinį palaikymą, numatome, kad

„Mobicents“ tai įgyvendina. Taigi našumo pagerėjimui pamatyti turėsime modifikuoti kodą, kad jis veiktų su 1.1 platforma bei komponentais.

Našumo pagerėjimas turėtų įtakoti tik per sekundę apdorojamų skambučių kiekį. Paslaugos spartai svarbiausią įtaką daro mūsų sukurti komponentai bei jų naudojami resursai. Paslaugos vidiniai skaičiavimai visiškai nepriklauso nuo JSLEE ar išorinių resursų. Duomenys yra gaunami tiesiogiai bendraujant su įvykio kontekstu. Dauguma resursų adapterių kalba tik įvykių pagalba. Tai reikalinga, nes įvykiai apie besikeičiančią resurso būseną praneša ir tuo besidominčioms paslaugoms (SBB komponentams). Tačiau realizuota paslauga nori tik jai vienai įdomių duomenų – užklauskos į duomenų saugyklą (duomenų bazę). Realizacijoje, interfeisas su saugykla yra veiksmas (*anl. activity*). Plačiau apie naudojamą resursų adapterio ir SBB architektūrą galima paskaityti projekto skyriuje arba [23]. Taigi, nors ir resursų adapterio interfeisas keičiasi, SBB su juo bendraus taip pat – ne per SLEE, o tiesiogiai.

4.3. Duomenų kiekio įtakos įvertinimas

Jau trumpai užsiminėme prieš tai buvusiam skyrelyje, kad pagrindiniai paslaugos spartos faktoriai priklausantys ne nuo tinklo ar platformos yra vidiniai skaičiavimai bei kreipimaisi į išorinius resursus.

Vidiniai skaičiavimai nėra sudėtingi. Optimizacija įmanoma, tačiau ji gali atnešti klaidų, pasunkėti palaikymas, kodo skaitomumas, o gaunamas pagerėjimas nepastebimas.

Visai kitai yra su išoriniais resursais. Kaip aiškiai matyti 13 pav. Išdėstymo vaizdas ir 14 pav. JSLEE specifikacijos architektūra, paslauga bendrauja su trimis išoriniais resursais: SIP telefonu, Diameter programa bei duomenų baze. SIP telefonas nėra įtakojantis paslaugos spartą. Kaip tik, gautas SIP įvykis yra atskaitos taškas nuo kada paslauga atlieka numatytus skaičiavimus. Įvykių šaltinis yra vartotojas.

Diameter apmokėjimo programa gali įtakoti paslaugos spartą. Tiksliau svarbu Diameter resursų adapterio bazinės operacijos: įvykių apdorojimas, veiksmo sukūrimas, prisiregistravimas prie jo ir pan. Šios funkcijos yra dažnai naudojamos ir yra gerai ištestuojamos prieš naujos komponento versijos platinimą. Apmokėjimo paslauga skaičiavimus gali atlikti ilgai, tačiau jei ji veikia kitame kompiuteryje ar naudoja mažai kompiuterio resursų – paslaugos spartai tai įtakos nedaro. Dabartinėje versijoje paslaugos apmokestinimas yra vykdomas lygiagrečiai su paslaugos teikimu: nelaukiama sėkmingo apmokėjimo atsakymo. Vienintelis galimas trukdis – žinutės suformavimas ir jos išsiuntimas. Netgi ir modifikavus paslaugą, kad ji būtų teikiama tik gavus teigiamą apmokėjimo atsakymą,

VPN paslaugos kūrėjui yra maža tikimybė padėti pagerinti. Kaip jau minėta, JSLEE viduje taisyti nėra ką (nebent yra klaida pačiam „Mobicents“ ar jo Diameter resursų adapterio kode). Sekantys kaltininkai gali būti tinklas (administratoriaus atsakomybė) ar pati apmokėjimo paslauga. Pastaruoju atveju, tai dažniausiai kritinė operatoriaus aplikacija, kurios modifikavimas ar netgi „pamatymas“ yra neįmanomas.

Paskutinis ir svarbiausias spartos atžvilgiu išorinis resursas yra duomenų bazė. Priklausomai nuo paslaugos naudojimo masto (vidinis kompanijos tinklas, viešas operatoriaus tinklas...), išdėstymo tai gali būti svarbus faktorius. Kiekvieno pokalbio pradžioje atliekamos mažiausiai 1 (iš 3 lentelių, kai paslauga neteikiama) daugiausiai 2 (iš visų 5 lentelių, kai paslauga teikiama) užklausos. Žvelgiant į 15 pav. Sistemos naudojamos duomenų bazės diagrama, galime numatyti jog didžiausios lentelės bus vartotojų informacija (sip_user), vartotojo apmokėjimo informacija (user_bill) bei vartotojų trumpi numeriai (extension). Šių lentelių kiekis yra tiesiogiai susijęs su paslaugos ar netgi operatoriaus vartotojų kiekiu. Kitose lentelėse talpinama informacija lyginant su jau minėta vartotojų informacija bus nedidelė. Kad lengviau įsivaizduoti duomenų kiekius ir santykius tarp lentelių įrašų kiekio, pateikiame galimą jų užpildymą pas vidutinio dydžio operatorių.

5 lentelė: Galimas VPN duomenų bazės užpildymas

Lentelė	Apibūdinimas	Įrašų kiekis
sip_user	Vartotojo (telefono) informacija	100000
extension	Trumpas numeris susietas su vartotoju	1000000
domain	VPBX – uždaras ratas vartotojų (miestas, rajonas, organizacija...)	10
user_bill	Vartotojo paslaugos apmokėjimo būdas	100000
payment	Galimi apmokėjimai	20

Dirbant su tokiais informacijos kiekiais, įtaka spartai bus jaučiama. Ypač, kai vykdymo laikas matuojamas sekundės dalimis. Pirmiausia reikėtų tiksliau įvertinti duomenų kiekio padidėjimo įtaką paslaugos spartai. Tolimesni patobulinimai priklausys nuo rezultatų. Pagrindinės galimybės yra: pagerinti aparatinę įrangą, naudoti minėtus JSLEE profilius, pakeisti duomenų bazę (kitas gamintojas, tipas...).

Pirmas atvejis yra lengviausias, tačiau gali būti jog vienintelis įmanomas ar netgi pigiausias. Neužtenka pagreitinti procesorių ar padidinti atmintį, galima žiūrėti į kitus tobulinimus: SSD naudojimas vietoj HDD, serverių kiekio padidinimas...

Apie profilius jau užsiminta kalbant apie JSLEE 1.1 naujoves. Tai tarsi žodynas kuriame saugoma rakto => reikšmės poros. Paieška atliekama pagal raktą ir tam tikrus kriterijus. Šis būdas yra naudingas, nes duomenys yra saugojami atmintyje, greitai pasiekiami. Tačiau trūkumai atsveria naudą: veiks tik tam kompiuteryje, kur veikia JSLEE, neįmanoma (ar bent sudėtinga) jų modifikuoti veikimo metu. Pastarasis trūkumas užkerta jų panaudojimą sukurtoje paslaugoje. Vienas iš reikalavimų yra galimybė vartotojui konfigūruoti paslaugos teikimą per grafinę sąsają.

Duomenų bazės, o ypač jos tipo, pakeitimas gali atnešti naudos, tačiau tam visų pirma reikia įsitikinti problemos egzistavimu ir svarbumu. Be to, tai yra ganėtinai platus darbas ir per daug nuklystama už atliekamo tyrimo ribų.

5. EKSPERIMENTINĖ DALIS

Pabandysime realizuoti skyriuje 4. Paslaugos tobulinimo galimybės aptartus paslaugos patobulinimus. Pradžiai supažindinsime su matavimo aplinka bei metodika. Vėliau pateiksime rezultatus.

5.1. Matavimo aplinka

Svarbos, per sekundę apdorojamų skambučių kiekio nustatymo matavimams buvo naudojama modifikuotas Mobicents testavimo komponentas (SBB) bei Sipp programa. Skirtingoms JSLEE versijoms yra naudojamas skirtingas testavimo SBB. Jo tikslas yra pagal numatytą scenarijų reaguoti į gautus, sugeneruoti naujus SIP įvykius.

Sipp – atviro kodo SIP protokolo testavimo įrankis / srauto generatorius. Jo įvedimo duomenys yra testavimo scenarijus. Scenarijuje aprašoma kokie SIP įvykiai kam ir kada yra generuojami, kaip reaguojama į atsakymą. Štai taip atrodo testo scenarijus:

```
INVITE ----->
 100 <-----
 180 <-----
 200 <-----
 Pauzė [ 300ms]
 ACK ----->
 Pauzė [ 55.0s]
 BYE <-----
 200 ----->
 Pauzė [ 1000ms]
```

Kairėje yra Sipp komponentas, dešinėje JSLEE. Rodyklės parodo įvykių keliavimo kryptį. Sipp įvykiai yra aprašomi tekstu (INVITE, ACK, BYE), o gaunami – identifikuojami kodu. Pauzė simuliuoja tinklo uždelsimus, skaičiavimus bei patį pokalbį (55s).

Aplinkos veiksniai įtakojantys rezultatus:

- Visi komponentai, išoriniai JSLEE resursai veikia tam pačiam kompiuteryje
- Abi testuojamos „Mobicents“ versijos naudojama kur galima vienodą konfigūraciją
- Testavimo metu yra išjungtas informacinis žurnalo pildymas
- Statistikai surinkti kas sekundę yra atnaujinami Sipp bei vmstat rezultatų failai

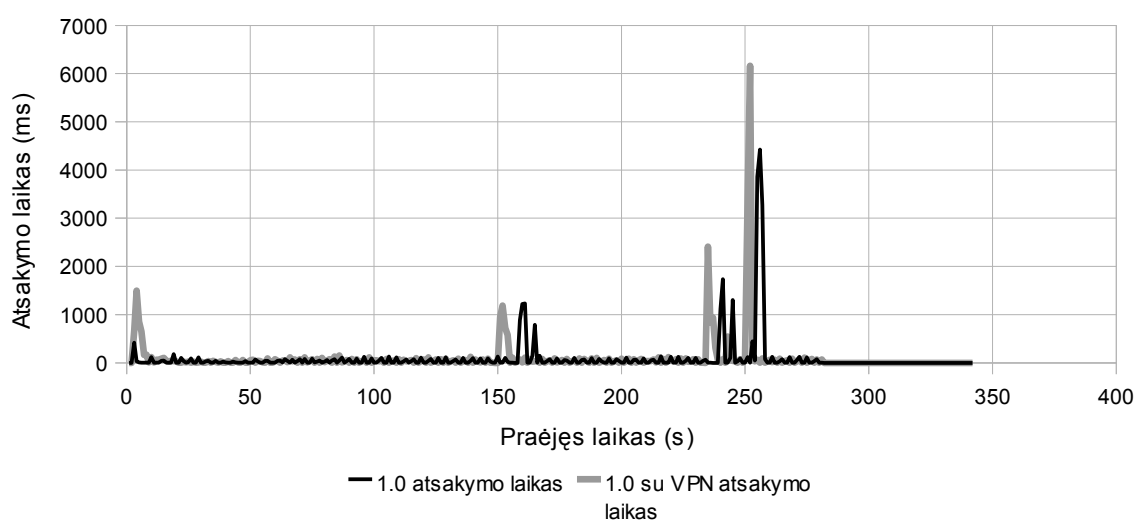
Sipp paleidimo duomenyse nurodomas generuojamų skambučių keikis bei dažnis.

Norint palyginti rezultatus stengtasi skambučių kiekį reguliuoti pagal dažnį, kad pilnas testas truktų ~ 5min. Skambučių kiekis per sekundę visuomet yra kuo didesnis.

Matavimų aplinkos, jos problemų suvokimas leidžia į tai atsižvelgiant įvertinti gautus rezultatus.

5.2. Paslaugos įtakos platformai įvertinimas

Paslaugos įtaką platformai įvertinsime peržiūrėdami Sipp užfiksuotas periodinius atsakymo laikus. Šis matas parodo per kiek laiko buvo atsakyta į išsiųstą įvykį. Taip pat galime įsivertinti ar paslauga vykdoma per užsibrėžtą 0.02 sekundžių laiko tarpą.



16 pav. 1.0 JSLEE platformos atsakymo laikas

Pradžiai nustatykime 1.0 platformoje veikiančios VPN paslaugos įtaką. Iš 16 grafiko matyti, jog atsakymo laikai yra panašūs. Turime keletą stiprių nukrypimų, kurių priežastis gali būti resursų trūkumas ar nepasidalinimas su kitomis sistemos operacijomis.

Iš pateikto grafiko sunku nustatyti vyraujančią vidurkį. Pasižiūrėjus į sugeneruotus duomenis matyti, kad reikšmės kinta tarp 1 ir 900 ms tiek veikiant paslaugai, tiek – ne. Taigi paslaugos vykdymo laikas gali viršyti arba nepasiekti užsibrėžto.

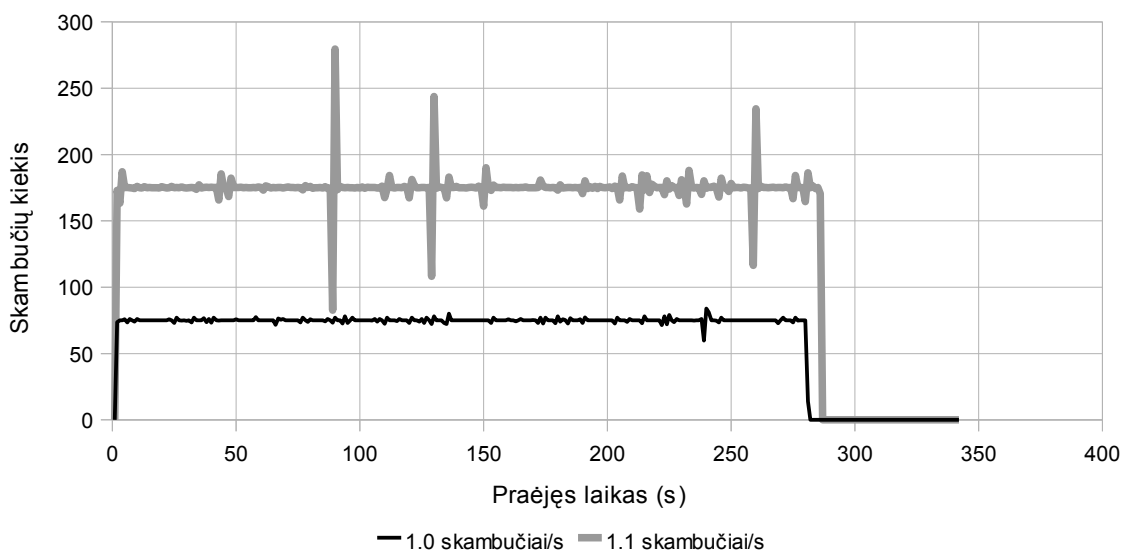
Vidurkis yra 103,37 – 1.0 versijoje be paslaugos ir 114,24 – 1.0 versijoje su VPN paslauga. Neleidžia vienareikšmiškai teigti, jog paslauga tenkina jai keliamus spartos reikalavimus, nes kaip matyti iš grafiko, kelios atsakymo laiko reikšmės net keliasdešimt kartų viršija užsibrėžtą. Reiktų atsižvelgti ir į tai, kad matavimo metu, kai buvo renkami duomenys, stengtasi nustatyti kuo didesnę apdorojamų skambučių skaičių taip apkraunant sistemą. Šis faktorius be abejo įtakojo ir matomus rezultatus.

Ką vidurkis parodo, tai kad sukurta paslauga įtakoja atsakymo išsiuntimo spartą. Sulėtėjimas sudaro apie 10% atsakymo išsiuntimo vidurkio be paslaugos (panašiai ir 1.1 versijoje). Šį faktą lemia tai, kad paslaugos teikimas reikalauja užblokuoti kiekvieno pradinio skambučio atsakymo siuntimą vartotojui, kol nebus išsiaiškintos paslaugos teikimo aplinkybės. Vidurkio su veikiančia ir be skirtumas parodo vidutinę paslaugos teikimo trukmę – 10,87. Tai tenkina spartos reikalavimą.

Kaip jau minėta, atitinkami rezultatai gauti ir 1.1 platformoje. Tik čia trukdžiai pastebėti dažnesni bet mažesni.

5.3. Platformos versijos pakeitimo įvertinimas

Šiame tyrime įeina tiek naujos „Mobicents“ tiek JSLEE specifikacijos versijos našumo įvertinimas. Šių faktorių atskirti neišeina, nes tik nuo naujos „Mobicets“ versijos palaikoma nauja specifikacija, o našumo paspartinimas taip pat įgyvendintas tik kartu su ja. Nors specifikacija įtakos tam neturėjo turėti, tačiau jos reikalauti pakeitimai turbūt leido platformos kūrėjams pataisyti ir patobulinti kitas vietas be atgalinio palaikymo baimės. Nauja versija nenutraukia 1.0 komponentų palaikymo, tačiau juos reikia šiek tiek modifikuoti. Ką kūrėjas gali nuspręsti atlikti įvertinęs teikiamus privalumus.

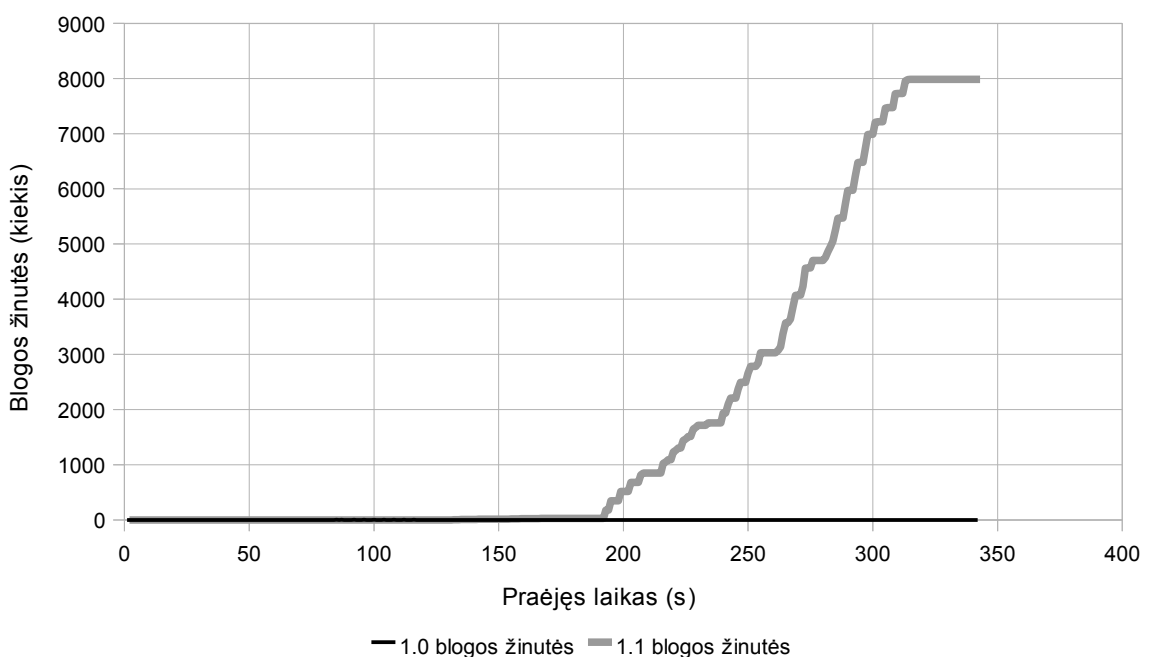


17 pav. Platformoje apdorojamų skambučių keikis per sekundę

Generuojamų skambučių kiekis yra nurodomas paleidžiant srauto generatorių Sipp. Maksimalus platformos palaikomų skambučių kiekis buvo nustatomas per naują paleidžiant šį matavimo įrankį, keičiant įėjimo reikšmes ir stebint, kad numatyto kiekio skambučių apdorojimo metu neatsirastų klaidų. Jei klaida yra, skambučių kiekis sumažinamas ir įrankis perleidžiamas.

Tokiu būdu nustatyta, kad 1.0 palaiko 75, o 1.1 – 175 skambučius per sekundę. Tai daugiau nei dvigubas pagerėjimas ir kol tai nėra toks išpūdingas kaip „Mobicents“ minimas 4 kartų pagerėjimas [20], mūsų atlikti matavimai yra tikslesni, nes abiem atvejais naudotos vienodos konfigūracijos bei aparatinė įranga. Tokie pat skaičiai gauti ir išmatavus abi versijas su veikiančia realizuota paslauga. Tai leidžia daryti prielaidą, kad realizuotos paslaugos įtaka platformai yra netokia žymi.

Pažvelgus į 17 grafiką be skirtumų tarp skambučių kiekio apdorojimo, į akis krenta ir pastebimai nestabilus jų generavimas 1.1 versijoje. Ar tik nebus taip, kad dėl spartos yra paaukojamas stabilumas? Ar šie nesklaidumai tik dėl padidėjusio apdorojamos informacijos kiekio (nesusidorojo aparatūra)? Bandydami atsakyti į šiuos ir panašius klausimus pasižiūrėkime į sekančius grafikus.

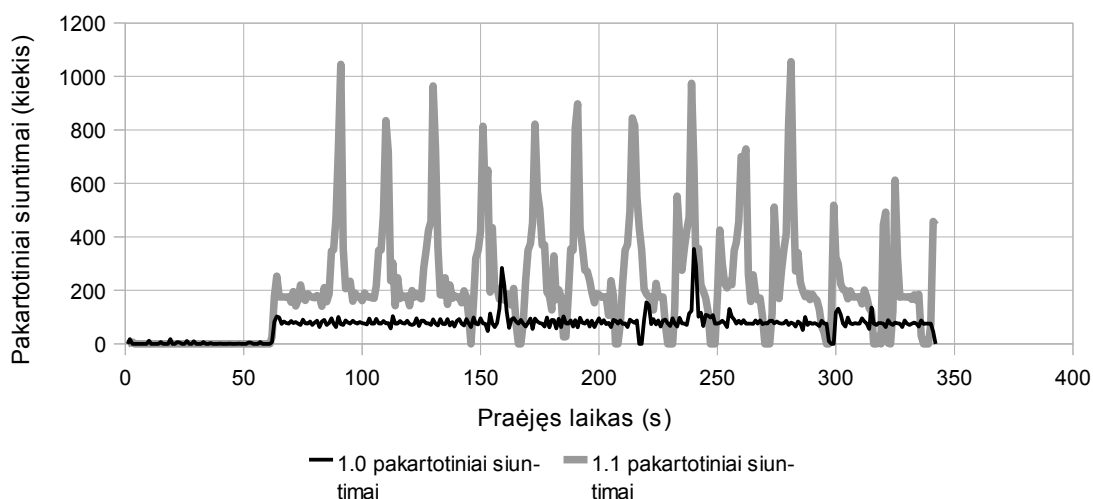


18 pav. Blogų žinučių kiekis pagal platformą

18 grafikas rodo gana prastą padėtį naujoje versijoje. Prieš tolesnius vertinimus svarbu paminėti, kad:

- blogos žinutės – tai Sipp gautos žinutės, kurios negali būti susietos su egzistuojančiu skambučiu (jis jau pašalintas) ar kurios kitaip neatitinka nurodytą scenarijų, dabartinę būseną, tačiau neįtakoja tolesnio egzistuojančių skambučių eigos (nėra klaida).
- generuojamų skambučių keikis skiriasi daugiau nei du kartus (1.0 – 21000, 1.1 – 50000). Taip yra dėl to, kad atlikti panašios trukmės testus. Grafike matyti, kad blogų žinučių kiekis padidėja praėjus daugiau nei pusei viso matavimo laiko. Tai reiškia kad apdorota daugiau nei 21000 skambučių ir problema gali būti nesusijusi su platforma, o su sistema.
- Apkrovimas. Per sekundę generuojamų skambučių kiekis buvo keičiamas 5. To pasekoje 1.1 versija gali būti arčiau kritinės ribos.

Taigi matome, kad vien šio grafiko duomenų išvadoms priimti neužteks. Pakartojus testavimus kelis kartus 1.0 versijoje blogų žinučių kiekis nepakyla aukščiau 100, o 1.1 – nenukrenta žemiau 1000.

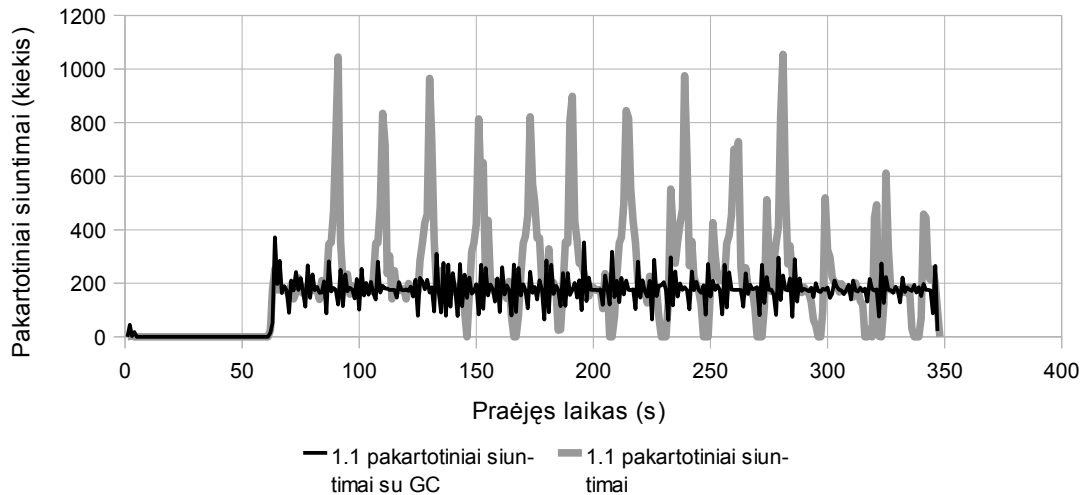


19 pav. Pakartotinis įvykių siuntimas pagal platformą

Pakartotinis įvykių siuntimas yra vykdomas jei Sipp negauna atsakymo per nustatytą laiko tarpą nuo pirminio įvykio išsiuntimo. 19 grafike nematyti mūsų problemą paaiškinančių ar įrodančių bruožų. Grafiko pakilimas ties ~60s sutampa su pirmais skambučių pabaigimais po pokalbio (toks Sipp scenarijus), kas reiškia srauto padidėjimą ir pasunkėjusį platformos susidorojimą su juo. To reikia tikėtis, nes mes stengiamės kuo labiau apkrauti sistemą. 1.1 versijos įvykių siuntimo mastai didesni, nes sekundės tarpe apdorojamas didesnis kiekis skambučių.

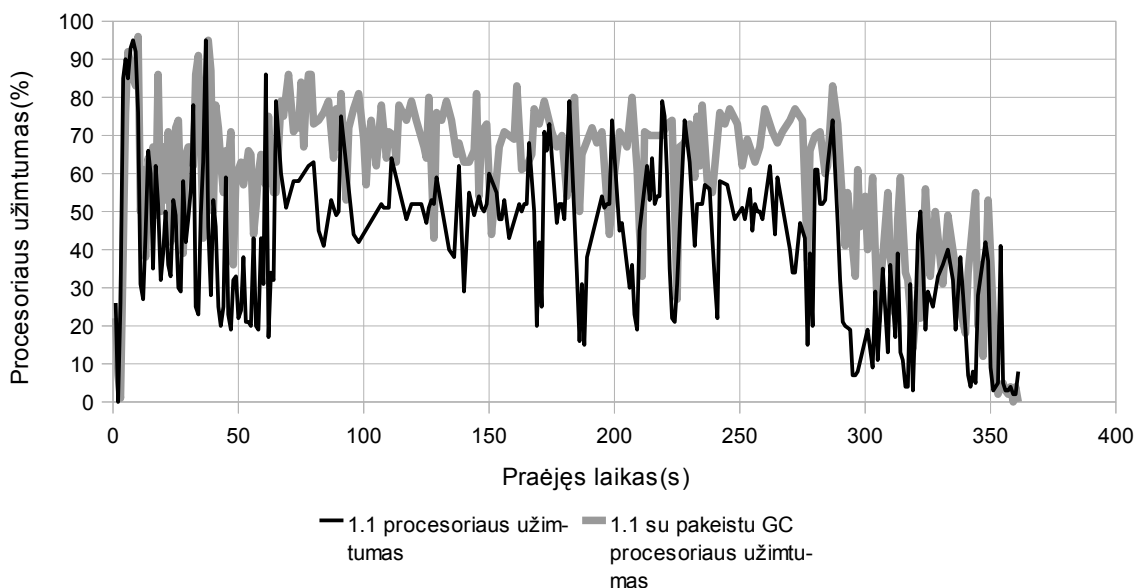
18 grafike pastebėtai anomalijai patvirtinti ar paneigti atlikome papildomus matavimus: sumažinus per sekundę generuojamų skambučių kiekį, bei nurodžius šiukšlių surinkėją (*angl. garbage collector*) (GC) tipą. Abiem atvejais blogų žinučių kiekis nukrito iki

0. Numatytas GC tipas turi ilgus poilsio tarpus, kurių metu blokuojamas JVM darbas. Įjungus padidinančio tipo GC, mes duodame didesnę darbą procesoriui, tačiau gauname pagerėjimą. Rezultatus galime pamatyti 20 pav. Šio tipo GC šiukšlių valymas atliekamas nestabdant JVM darbo ir yra naudingas 1-2 branduolių sistemose. GC vykdymo metu reikalingi veiksmai yra suskaidomi ir paleidžiami trumpų pauzių metu.



20 pav. Padarytų pakartotinių siuntimų palyginimas naudojant skirtingus GC

20 grafike pavaizduoti pakartotiniai siuntimai su skirtingo tipo GC veikiant 1.1 VPN paslaugai. Iš jo matyti, kad su nauju GC pakartotinių siuntimų kiekis yra stabilus. Su seno tipo GC vidutiniškai išsiunčiami 209,9 pakartotiniai siuntimai. Su nauju GC pakartotinių siuntimų kiekis nukrenta iki 146,57.



21 pav. Procesoriaus sąnaudų įvertinimas su skirtingo tipo GC

Kaip jau užsiminta anksčiau, bei oficialioje Sun Java dokumentacijoje, naujo tipo GC naudoja daugiau procesoriaus resursų. Tuo galime įsitikinti pažiūrėję į 21 grafiką. Jame

pateikiami vartotojo paleistų programų naudojamo procesoriaus procentai. Viso matavimo laikotarpyje, naujojo GC kreivė yra virš seno – jis naudoja daugiau procesoriaus resursų. Vidurkiai abiem atvejais yra gana žemi – 29,21% ir 39,17% atitinkamai seno ir naujo GC. Vidurkis įskaičiuoja ir paskutinį matavimų etapą kai skambučiai yra sugeneruoti, o laukiama jų pokalbio pabaigos – sąnaudos minimalūs. Grafike matome, kad didžiausios apkrovos metu, senas GC sunaudoja >50% procesoriaus resursų su vis pasikartojančiais ryškiais sąnaudų padidėjimais. Tuo tarpu naujas GC – sunaudoja ~70% procesoriaus resursų, tačiau sąnaudų padidėjimas mažesnis.

Nustatėme, kad aptiktą problemą galime spręsti keliais būdais: srauto sumažinimu (mažesnis per sekundę apdorojamų skambučių kiekis) arba pakeisdami JVM GC tipą. Antru atveju padidėja procesoriaus resursai. Susidūrus su problema, atsakingas asmuo turėtų įvertinti sąlygas ir aplinkybes bei priimti tinkamą sprendimą. Šis sprendimas priklausys nuo veikimo konteksto.

5.4. Duomenų bazės įtakos paslaugos našumui įvertinimas

Matuojant duomenų bazės įtaką pagal 5 lentelę buvo sugeneruota ir užpildyta testinė duomenų bazė. Sugeneruoti duomenys nebuvo naudojami Sipp scenarijui, tačiau užklausos metu jie buvo perrenkami ir turėjo simuliuoti duomenų atrinkimą didelėje duomenų bazėje.

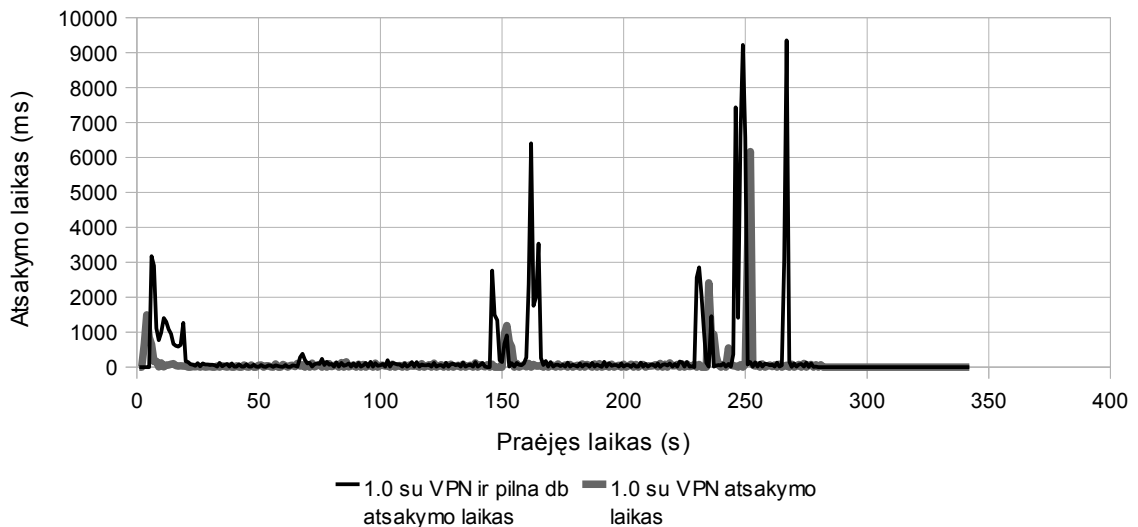
Rezultatai buvo itin nepalankūs. Su pradiniu skambučių per sekundę kiekiu (75 – 1.0 bei 175 – 1.1), nesulaukta nei vieno sėkmingo skambučio pabaigos. Sumažinus skambučių kiekį iki 20, per 10 minučių pasibaigė tik 27 skambučiai (0 klaidų) iš 5000, nors visi matavimai turėjo vykti 5 minutes. Visą tą laiką procesoriaus apkrovimas viršijo 90%.

Matome, kad tokios apimties duomenų bazė testavimo aplinkoje dirbti negali. Testą reikėtų pakartoti duomenų bazę iškėlus į kitą kompiuterį, kad padidinti JSLEE prieinamų procesoriaus resursų kiekį. Nepaisant to, pagerėjimas gali būti nepakankamas ir prireikti kitokių patobulinimų.

Šioje duomenų bazėje užklausa, kuri yra vykdoma visų skambučių pradžioje, trunka 0,7 – 0,9 sekundės. Toks laikas užtrunka testavimo kompiuteryje gauti rezultatus pateiktai MySQL sql užklausiai. Aišku šiuo laikotarpiu stengiamasi išnaudoti kuo daugiau resursų taip kuo greičiau pateikiant rezultatą. Paslaugos SBB sustabdo tolesnį SIP įvykio apdorojimą JSLEE taip apkraudamas ir platformą.

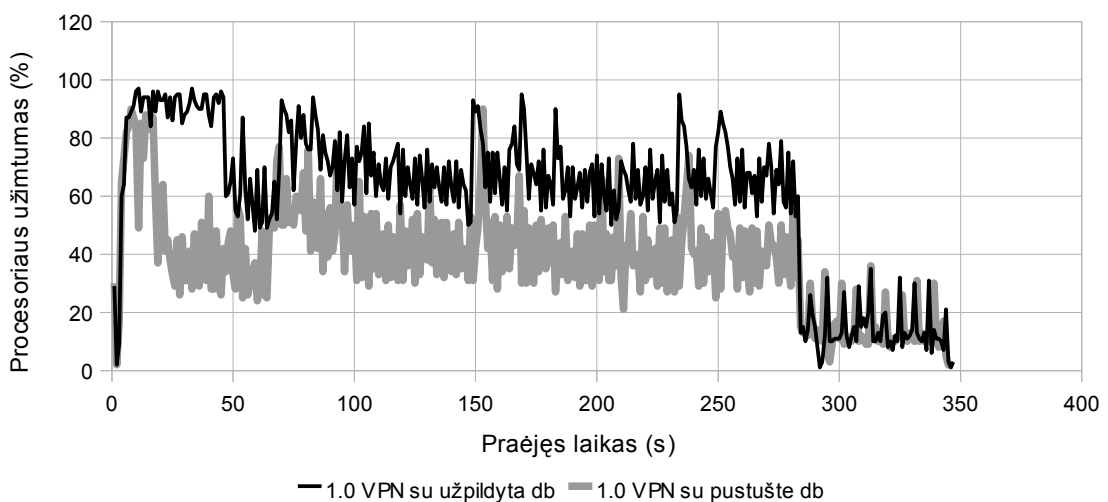
Pakartotinis matavimas atliktas su mažesniais duomenimis: vartotojų bei papildomų numerių skaičius sumažintas iki 10000. Šiuo atveju sql užklausa vykdoma 0,09 – 0,1

sekundės. Paleidus Sipp srauto generatorių su naudotais kai duomenų bazė buvo pustuštė, matome, kad rezultatai yra šiek tiek prastesni (1.0 versija su 75 skambučių per sekundę generavimu aptiko 22 klaidas), tačiau palyginami.



22 pav. 1.0 versijos VPN atsakymo laiko santykis pagal duomenų bazės pilnumą

22 grafike palyginimui su pilna duomenų baze naudojami tie patys duomenys kaip ir 16 grafike. Kaip buvo matyti pastarajame, paslauga suteikia tam tikrą uždelimą. Duomenų bazės užpildymas taip pat šį delsimą padidina. „Mobicents“ testavimo paslaugos atsakymų vidurkis yra 103,37, o 114,24 – 1.0 versijoje su VPN paslauga, kai duomenų bazė yra ne tuščia – 396,16. Šiuo atveju pokytis ryškus. Grafike tai ryškiai nesimato. Tačiau tokio rezultato reikėjo tikėtis, kadangi jau minėta, jog šis rezultatas sėkmingai nepasibaigė – aptikta 22 kritinės klaidos. Nepaisant to, pokytis nėra labai didelis ir lengvai ištaisomas aparatinį resursų pagerinimu.



23 pav. Procesoriaus užimtumas pagal duomenų bazės užpildymą

a

Pastarasis, penkiaženklės duomenų bazės įrašų nesiekiantis, matavimo atvejais gali būti netgi realesnis. Organizacijos aptarnaujančios didesnius vartotojų kiekius naudoja daug kartų našesnę aparatinę įrangą ir tokio atvejo testavimas ant vieno kompiuterio nėra labai naudingas. Nepaisant to, įsitikinome, kad duomenų saugojimas ir išgavimas turi labai svarbią reikšmę. Paslaugos ar jų grupės projektuotojas turi numatyti aptarnaujamų vartotojų kiekį ir atitinkamai reaguoti.

6. IŠVADOS

Naujų technologijų atsiradimas ir ypač interneto plėtra verčia telekomunikacijų bendroves atsinaujinti. Šiuo metu dauguma pilnai ar dalinai migruoja į NGIN paslaugas ir IMS paremtą architektūrą. Ši architektūra remiasi internetu kaip transporto sluoksniu. Esamos telekomunikacijų architektūros sujungimas su internetu atveria gausybę galimybių. Sujungiant sparčiai kuriamas internetines paslaugas su telekomunikacijų paslaugomis bei kuriant anksčiau neįmanomas paslaugas vartotojui suteikiama patirtis atneša didesnę pelną.

Darbo metu susipažinta su telekomunikacijų sritimi, naujos architektūros naujovėmis bei realizacijos galimybėmis. Susipažinta su populiariausiomis SIP protokolo platformomis; SIP Servlets bei JSLEE. Jos įvertintos ir pasirinkta labiausiai priimtina „Mobicents“ JSLEE specifikacijos platforma.

Pasirinktoje platformoje realizuota veikianti trumpų numerių paslauga (VPN). Kadangi realizuotos sistemos sritis – nauja, naudojami terminai bei veikimo aplinka – taip pat, daug laiko užtruko reikalingos informacijos išsiaiškinimas. Nuspręsta, kad paslaugos logikos komponentas (SBB) su duomenų baze resursų adapterio pagalba bendraus tiesiogiai, ne įvykiais. Pastarieji tik padidintų apkrovą neteikdami jokios naudos. Programa veikia pagal nustatytus reikalavimus ir buvo pristatyta užsakovui.

Užsibrėžėme kelis kriterijus pagal, kuriuos vertinome paslaugos bei platformos darbą ar patobulimus: sparta, per sekundę apdorojamų skambučių kiekis, įtaka aparatinei įrangai bei platformos pasirengimas plačiam naudojimui. Buvo numatyta, kad kriterijams turės įtakos: JSLEE specifikacija, platformos versija, duomenų saugojimo būdas. Detaliau panagrinėjus šių faktorių svarbą, nuspręsta, kad nauja platformos versija turės didžiausią teigiamą įtaką.

Ekspperimentų metu bandyta paneigti ar patvirtinti tiriamosios dalies teorinius pasvarstymus ir sprendimus. Tam reikėjo realizuotą paslaugą pritaikyti JSLEE 1.1 specifikacijai. Ši užduotis tapo sudėtingesnė nei iš pradžių buvo manyta. Pagrindinės to

priežastys: konfigūracijos pasikeitimai, kurie neaptinkami kompiliavimo metu, o juo aprašantys klaidų pranešimai painūs.

Ekperimentiniai matavimai parodė, kad iš tikrųjų – nauja „Mobicents“ versija ne tik veikia pagal JSLEE 1.1 specifikaciją, bet yra ir spartesnė. Pastebėtas ~233% per sekundę apdorojamų skambučių skaičiaus pagerėjimas. Šio matavimo metu pastebėtas didelis gaunamų klaidingų žinučių skirtumas tarp platformos versijų. Atlikus detalesnę analizę ir eksperimentus, nustatyta, kad to priežastis – arti kritinės ribos (kai nebespėjama apdoroti žinučių) esantis srautas. Pasiūlyti du sprendimai: sumažinti srautą arba pakeisti naudojamą JVM šiukšlių surinkėjo tipą.

Nustatyta, kad duomenų saugojimas turi didesnę įtaką nei tikėtasi. Esant dideliame kiekiui duomenų paslaugos teikimas naudotoje eksperimentinėje aplinkoje yra neįmanomas. Po detalesnės analizės pasiūlyta diegiant įvertinti naudojamus duomenis bei išorinių komponentų išdėstymą.

7. LITERATŪRA

- 1 Lietuvos respublikos ryšių reguliavimo tarnyba, Elektroninių ryšių paslaugų kainų palyginimo sistema - [žiūrėta 2009-01-12]. Prieiga per internetą: <http://www.skaiciuok.lt>.
- 2 Statistikos departamentas prie Lietuvos respublikos vyriausybės [žiūrėta 2010-05-24]. Prieiga per internetą: <http://www.stat.gov.lt/>
- 3 J. Zuidweg, Next Generation Intelligent Networks, Artech House Publishers, 2001.
- 4 Pasaulinė interneto naudojimo statistika - [žiūrėta 2008-10-20]. Prieiga per internetą: <http://www.internetworldstats.com/stats.htm>, Atnaujinta: 2008 liepos 30 dieną.
- 5 UMTS Forumas, „TOP 3 3G paslaugų rankas ,” UMTS forumas, Tech. Ats., 2001.
- 6 N. Galea, „10 priežasčių pereiti į IP PBX“, 2007 - [žiūrėta 2008-11-13]. Prieiga per internetą: <http://www.3cx.com>
- 7 M. Tadault, S. Soormally, L. Thiébaut, Tinklo evoliucija link IP multimedijos, 2004 - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www1.alcatel-lucent.com/doctypes/articlepaperlibrary/pdf/ATR2003Q4/T0312-IP-Multimedia-EN.pdf>
- 8 Tarptautinis inžinerijos konsorciumas: Web ProForums, Sumanus tinklas - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.iec.org/online/tutorials/in>
- 9 R. LJ Chen, E. CY Su, V SC Shen, Y. Wang, Įžanga į IP multimedijos posistemį (IMS), Pirma dalis SOA Parlay X Žiniatinklio paslaugos, 2006 11 12 - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.ibm.com/developerworks/webservices/library/ws-soa-ipmultisub1/>
- 10 Ericsson Aiškinamasis dokumentas, IMS ekosistemos paslaugos, 2007 02 - [žiūrėta 2010-05-21]. Prieiga per internetą: http://www.ericsson.com/technology/whitepapers/3109_Services_in_the_IMS_ecosystem_A.pdf
- 11 Cisco Sistemų aiškinamasis dokumentas, Operatoriaus lygio naujos kartos intelektualiujų paslaugų tinklo kūrimas, 2006 - [žiūrėta 2010-05-21]. Prieiga per internetą: http://www.cisco.com/en/US/prod/collateral/routers/ps5763/prod_white_paper0900aecd802e2a52.pdf

- 12 M. Tadault, S. Soormally, L. Thiébaud, Tinklo evoliucija link IP multimedijos, 2004 - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www1.alcatel-lucent.com/doctypes/articlepaperlibrary/pdf/ATR2003Q4/T0312-IP-Multimedia-EN.pdf>
- 13 JAINSLLEE.ORG, JAIN SLEE ir susijusios technologijos - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.jainstee.org/othertechnologies/other.html>
- 14 JAIN SLEE and SIP Servlet, Jslee.org - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.jainstee.org/othertechnologies/sleevservlet.html>
- 15 Rhino Documentation, Comparison SIP Servlet - [žiūrėta 2010-05-21]. Prieiga per internetą: <https://developer.opencloud.com/devportal/display/RD/2.2+SIP+Servlet>
- 16 JSR 22: JAIN SLEE (JSLEE) 1.0 Specification, 2004 - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.jcp.org/en/jsr/detail?id=22>
- 17 JSR 240: JAIN SLEE (JSLEE) 1.1 Specification, 2008 - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.jcp.org/en/jsr/detail?id=240>
- 18 jNetX Smashes Java XTP Performance Records for High Throughput and Low Latency, 2007 – [žiūrėta 2010-05-21]. Prieiga per internetą: http://www.jnetx.com/index.php?id=139&no_cache=1&file=55&uid=1424
- 19 Rhino Scalability and Performance on Blade hardware, 2007– [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.opencloud.com/documents/Rhino%20Performance%20on%20Blade.pdf>
- 20 Mobicents performance guide - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://groups.google.com/group/mobicents-public/web/mobicents-performance-guide>
- 21 SUN's UltraSparc T1 - the Next Generation Server CPUs , 2005 12 19 - [žiūrėta 2010-05-21]. Prieiga per internetą: <http://www.anandtech.com/show/1909>
- 22 Differences between JAIN SLEE versions 1.0 and 1.1 - [žiūrėta 2010-05-22]. Prieiga per internetą: <https://developer.opencloud.com/devportal/display/RD2v0/1.4.2+Scope+of+the+JAIN+SLEE+Specification>
- 23 Mobicents Persistence RA design - [žiūrėta 2009-08-30]. Prieiga per internetą: <http://groups.google.com/group/mobicents-public/web/persistence-ra-design>

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

3G – telekomunikacijų standartas, 3G tinklas tai mobiliojo ryšio tinklas, kuris gali perduoti didelės spartos internetą ir video telefoniją.

API (angl. *Application programming interface*) - aplikacijos programavimo interfeisas, funkcijų ir bibliotekų rinkinys, kuris leidžia programai tiesiogiai bendrauti su operacine sistema.

Aplikacijų serveris (IMS architektūra) – programinė platforma atsakinga už efektyvų paslaugų vykdymą bei saugojimą.

CDMA – telekomunikacijose tai CDMA2000 ir cdmaOne standartų sutrumpinimas.

CDMA2000 – mobiliojo ryšio technologija, naudojama balso, informacijos ir signalų informacijos siuntimui tarp mobilių telefonų ir stočių.

Diameter – kompiuterinių tinklų protokolas, skirtas prisijungimui, autorizacijai ir apskaitai (**RADIUS** protokolo pakaita).

GC (angl. *garbage collector*) – šiukšlių surinkėjas.

GPRS – mobiliojo ryšio technologija, skirta duomenų perdavimui GSM ir D-AMPS tinkluose

GSM (angl. *Global Standart for Mobile Communications*) – globalaus mobilių telefonų ryšio standartas.

HSS (angl. *Home subscriber server*) (IMS architektūra) – pagrindinė vartotojų duomenų bazė, kuri susikalba su IMS tinklo nariais, kurie valdo skambučius.

I-mode – bevielio prisijungimo prie interneto paslauga Japonijoje.

IMS (angl. *IP Multimedia Subsystem*) – IP multimedijos posistemis, tai architektūra pateikti IP multimedijos paslaugas.

IN (angl. *Intelligent network*) - nepriklausomų paslaugų telekomunikacijų tinklas, tinklo architektūra fiksuotiems ir mobiliems telekomunikacijų tinklams. **NGIN** – naujesnis šio tinklo atitikmuo.

Internetinės paslaugos (angl. *Web Services*) – programų sistema sukurta tarpmašininiam bendravimui interneto protokolu.

IP – interneto protokolas, jo pagalba internetu keliauja duomenys

Išieigos taškas (angl. *gateway*) – tinklo dalis pritaikyta bendravimui su kitu tinklu skirtingais protokolais.

JAIN (angl. *Java APIs for Integrated Networks*) – API kūrimas telefonijos (vaizdo ir garso) paslaugoms.

JSLEE - Java paslaugų vykdymo aplinka, Java SLEE standarto realizacija.

JVM (angl. *Java virtual machine*) – leidžia Java programavimo kalba parašytas programas vykdyti tam tikroje aplinkoje.

Laidinis tinklas – laidais į tinklą sujungti mazgai, terminalai. Duomenys keliauja laidais.

PBX (angl. *Private branch exchange*) – telefonų skirstytuvas veikiantis ne pas operatorių, o konkrečiame ofise ar firmoje.

Resursų adapteris (angl. *resource adaptor*) – JSLEE platformos komponentas atsakingas už tarpinikavimą tarp jos ir išorinių resursų.

Ro/Rf (IMS architektūra) – apmokėjimo interfeisai. Atitinkamai aktyvus (angl. *online*), kitaip kreditinis ar išankstinio apmokėjimo, ir neaktyvus (angl. *offline*), vartotojai paslaugas apmoka pastoviais intervalais (mėnesio gale), apmokėjimas.

SBB, *Paslaugos sudedamojo dalis* (angl. *Service Building Block*) – pagal JSLEE specifikaciją tai yra programinis komponentas gaunantis ir inicijuojantis įvykius bei pagal juos ir esamą būseną atliekantis skaičiavimus.

SDP (angl. *Service delivery platform*) – grupė komponentų (platforma) teikianti paslaugų valdymo architektūrą (jų sukūrimą, sesijos kontrolę, protokolus) tam tikro tipo paslaugoms.

SIP (angl. *Session initiation protocol*) – signalizavimo protokolas, naudojamas multimedijos komunikavimo sesijų internete sukūrimui ir užbaigimui.

SIP Servlets – SIP protokolo realizacija telekomunikacijos paslaugų teikimui.

SLEE – didelio pralaidumo, žemo gaišties laiko veiksmų apdorojimo aplikacijų aplinka

SS7 (angl. *Signaling System Number 7*) – aibė telefonijos signalizavimo protokolų naudojamų reguliuoti standartinio telefonų ryšio skambučius

Standartinė telefonija – telefonija teikianti balso paslaugas telefoniniais tinklais (telefonija nenaudojanti IP)

VoIP – pokalbių perdavimas interneto protokolu.

VPN - Virtualaus privataus numeravimo paslauga (*angl. Virtual Private Numbering*), telekomunikacijų paslauga leidžianti vartotojui susikurti trumpus numerius kuriais jis tampa pasiekiamas kitiems asmenims.

QoS – paslaugos kokybė (*angl. Quality of Service*)

WAP (*angl. Wireless application protocol*) – tarptautinis standartas sukurtas mažiems nešiojamiems įrenginiams priėjimo prie interneto suteikti galimybę.

WiFi – bevielio ryšio technologija

WiMAX (*angl. Worldwide interoperability microwave access*) – bevielio ryšio technologija, kuri leidžia sparčiai perduoti duomenis dideliais atstumais.

9. PRIEDAI

9.1. Matavimo aplinka, įrankiai bei konfigūracija

Matavimai atlikti su Sipp programa. Jos dokumentacijoje pabrėžta, kad Windows operacinės sistemos versija susidoroja su mažesniais duomenų kiekiais ne kitų operacinių sistemų atitikmenys. Tuo buvo įsitikinta ir matavimo metu – 1.0 ir 1.1 „Mobicents“ versijos Windows OS rezultatai buvo identiški. To pasekoje pasirinkta Linux operacinė sistema. Naudota OpenSuse 11.2 Linux distribucija.

Sipp iškvietimo komanda:

```
sipp 127.0.0.1:5060 -inf users.csv -trace_err -trace_rtt
-trace_screen -trace_stat -sf uac.xml -i 127.0.0.1 -p 5050 -r
65 -m 3000 -l 100000 -nd -fd 1
```

Pagrindinių parametrų, įvedimo duomenų paaiškinimai:

- `-inf users.csv` – scenarijuje dalyvaujančių SIP telefonų informacijos vieta. Sipp pasirenka iš keturių nurodytų numerių, iš kurių 1 matuojama paslauga neteikiama.
- `-trace_*` - nurodymai rinkti statistiką. Statistika pildoma matavimų eigove ir rašoma į failą CSV formatu
- `-r 75` – per sekundę generuojamų skambučių kiekis
- `-m 3000` – scenarijaus metu sugeneruotų skambučių kiekis. Pasiėkus šią reikšmę Sipp nustoja generuoti naujus skambučius. Pagal šio ir per sekundę generuojamų skambučių kiekio dydžių santykį buvo reguliuojama matavimo trukmė. Tikslas suvienodinti ją esant skirtingiems duomenų srautams

- -l 100000 – maksimalus kiekis lygiagrečiai vykdomų skambučių (kad neperkrauti sistemos)
- -nd – išjungti numatytą Sipp elgesį. Jis testui nenaudingas.
- -fd 1 – statistikos atnaujinimo, įrašymo į failą dažnumas sekundėmis.

„Mobicents“ paleidimo metu nustatomi JVM parametrai:

```

JAVA_OPTS="-Xms1000m -Xmx1000m -Xmn256m -XX:MaxPermSize=128m
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000"

```

Papildomas parametras, nustatantis kitą JVM šiukšlių surinkėjo tipą:

```
-XX:+CMSIncrementalMode -XX:+CMSIncrementalPacing
```

Šis parametras naudojamas tik vieno matavimo metu.

Sistemos resursų matavimams naudota linux komanda vmstat. Ji kas sekundę spausdino dabartinės sistemos resursų panaudojimą. Procesoriaus matavimo grafikams buvo naudojama vartotojo naudojama proceso dalis. Rezultatuose 100% procesoriaus susideda iš vartotojo paleistų programų (naudojama grafikams), operacinės sistemos vidinių funkcijų, nuskaitymo/įrašymo laukimo bei laisvo procesoriaus dažnių procentų.

SQL užklauso vykdomo trukmė yra gauta ją įvykdžius per mysql komandą.