

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Aktorių modelių grįstos programų sistemos našumo vertinimas

Performance evaluation of actor model based software

Magistro baigiamasis darbas

Atliko: Džiugas Pocius (parašas)

Darbo vadovas: lekt. Donatas Čiukšys (parašas)

Recenzentė: doc. dr. Audronė Lupeikienė (parašas)

Vilnius – 2017

Santrauka

Šiame darbe nagrinėjamas aktorių modeliu grįstų programų sistemų našumo vertinimas. Poreikis nustatyti sistemos našumą gali atsirasti siekiant iš anksto įvertinti planuojamų pakeitimų įtaką, vertinant pradinį sistemos projektą ar prognozuojant sistemos apkrovos pasikeitimus. Šiais atvejais reikalinga paprastai naudojama priemonė, kuri leistų greitai rasti našumo rezultatą.

Pirmiausia darbe aprašomas aktorių modelis, apžvelgiami našumo vertinimo metodai ir išrenkami tinkami aktorių modeliu grįstoms sistemoms. Toliau pateikiamas eilių teorijos metodo papildymo pasiūlymas, leidžiantis vertinti našumą sistemose su išsišakojančios komunikacijos šablonu. Pabaigoje pateikiami pasiūlymai ir pavyzdžiai, kaip naudoti našumo vertinimo metodus praktiškai.

Summary

In this paper performance evaluation of actor model base software is analyzed. The need to determine software performance comes when predicting performance impact of change requests, evaluating initial software design or expecting changes in system's load. In these cases simple, easy to use way to predict performance is needed, which would let find performance characteristics fast.

First of all in this work the actor model is described, performance evaluation methods are reviewed and chosen for the actor model based software. Next, extension to the queueing theory method is present, which allows to use method for systems with fork-join communication pattern. Last part of the work consists of suggestions and examples of using performance evaluation method in practice.

Turinys

ĮVADAS.....	4
1.KLASIKINĖS SISTEMOS.....	8
2.REAKTYVIOS SISTEMOS.....	9
3.SISTEMŲ NAŠUMO VERTINIMAS.....	10
4.ANALIZĖS BŪDAI.....	11
4.1. Simuliaciniai metodai.....	11
4.2. Analitiniai metodai.....	12
5.PRANEŠIMŲ PERDAVIMO SISTEMŲ NAŠUMO VERTINIMAS.....	13
6.NAŠUMO VERTINIMAS NAUDOJANT EILIŲ TEORIJA.....	15
7.AKTORIŲ MODELIS.....	16
8.NAŠUMO VERTINIMO MODELIAI.....	20
8.1. Eilių tinklas.....	20
8.2. Tikslioji vidutinių reikšmių analizė.....	22
8.2.1.Skaičiavimo pavyzdys.....	24
8.3. Apytikslė vidutinių reikšmių analizė.....	27
8.3.1.Skaičiavimo pavyzdys.....	28
8.4. Eilių teorija.....	31
8.4.1.Atviras eilių tinklas su išsišakojimais.....	33
8.4.2.Skaičiavimo pavyzdys.....	34
8.4.3.Skaičiavimo tikrinimas simuliacija.....	36
9.AKTORIŲ MODELIO IR NAŠUMO VERTINIMO MODELIŲ SUSIEJIMAS.....	41
10.NAŠUMO VERTINIMO MODELIŲ NAUDOJIMAS.....	42
10.1. Sistemos struktūros pakeitimų įtakos vertinimas.....	42
REZULTATAI IR IŠVADOS.....	46
ŠALTINIAI.....	47

ĮVADAS

Nagrinėjamas objektas

Pagrindinis šiame darbe nagrinėjamas objektas yra aktorių modeliu grįstų programų sistemų, kuriose komunikacija vykdoma asinchroniniais pranešimais, našumo charakteristikų, tokių kaip vidutinių eilių ilgių, atsako laiko, išnaudojimo (angl. utilization) ir kitų, skaičiavimas.

Aktualumas

Programinės įrangos kūrimo metu visuomet svarbu kuo anksčiau identifikuoti galimas problemas. Tai gali būti ir kokybinių charakteristikų atitikimo numatytoms reikšmėms tikrinimas. Gali būti tikrinamos tokios programinės įrangos kokybinės charakteristikos, kaip atsako laikas (angl. response time), pralaidumas (angl. throughput), delsa (angl. latency) ir kitas. Parengus specialų kuriamos sistemos modelį atsiranda galimybė įvertinti kokybinių charakteristikų reikšmes prieš realizaciją. Tokie modeliai dažniausiai yra sudaromi remiantis formalia arba dalinai formalia kalba, griežtai apibrėžiant elementų savybes, sąveiką tarp elementų ir kitą susijusią informaciją. Charakteristikos nustatomos analizės metodais, naudojant matematinės, statistinės priemones.

Nagrinėjant sistemą našumo atžvilgiu, kyla tokie klausimai kaip, koks yra visos sistemos atsako laikas, esant konkrečiai sistemos elementų struktūrai. Taip pat galima problema yra pakeitimo sistemoje įtakos įvertinimas našumo atžvilgiu. Tarkim, pridėjus papildomų elementų ar pakeitus esamą sistemos struktūrą, ar pagerės visos sistemos atsako laikas. Esant nustatytam nepakankamam sistemos našumui, tačiau neidentifikuojant konkrečių sistemos elementų lemiančių veikimo stabdymą, galima situacija, kai atliekami visos sistemos spartinimo pakeitimai, tačiau iš tiesų sistemoje užklausų apdorojimą stabdo tik tam tikri atskiri elementai. Toks spartinimas tiksliai neidentifikavus problemos priežasties yra ekonomiškai nenaudingas.

Kadangi šiuo metu kompiuterių spartos didinimas daugiausiai vyksta lygiagrečio atžvilgiu, pavyzdžiui didinant branduolių skaičių procesoriuose, populiarėjant klasteriams, klasikinės architektūros, kurios grįstos sinchronine komunikacija, tampa netinkamos siekiant pilnai išnaudoti visus prieinamus resursus. Kaip alternatyva klasikinės struktūros sistemoms siūloma asinchroniniais pranešimais grįstas sistemas, sudarytas iš atskirų, nepriklausomų elementų, atliekančių užklausų apdorojimą.

Komunikacija asinchroniniais pranešimais grįstos sistemos, sudarytos iš nepriklausomų užklausų apdorojimo žingsnius atliekančių elementų, našumo kokybinių charakteristikų

nustatymas, remiantis sudarytu sistemos modeliu, architektūros aprašu, leistų parinkti geriausią, teikiančią mažiausią užklausos apdorojimo laiką, sistemos elementų išdėstymo ir sujungimo būdą.

Problema

Darbe nagrinėjama problema, kai yra turima:

- komunikacija asinchroniniais pranešimais grįsta sistema,
- sudaryta iš atskirų, nepriklausomų, apdorojimo veiksmus vykdančių darbo elementų,
- pateikiamas darbo elementų sujungimas (komunikacijos keliai),
- pateikiamos atskirų darbo elementų kokybinės charakteristikos,

ir yra ieškomos visos tokios duotos sistemos, atskirų užklausų apdorojimo našumo charakteristikos.

Darbo tikslas

Pasiūlyti praktiškai pritaikomą metodą, kaip paskaičiuoti išskirstytos sistemos našumo charakteristikas: vidutinius eilių ilgius, atsako laiką, išnaudojimą (angl. utilization), taikant ribojimus:

- Sistema išreikšta aktorių modeliu
- Pranešimai į sistemą ateina atsitiktinai pagal statistinį (Puasono) skirstinį
- Aktorių darbo trukmė atsitiktinė pagal statistinį (Ekspontentinį) skirstinį

Darbo uždaviniai

1. Surasti ir atrinkti pagal apribojimus našumo skaičiavimo matematinius metodus
2. Praplėsti metodus išskirstytų sistemų skaičiavimui
3. Pateikti pasiūlymus kaip praktiškai naudotis skaičiavimais

Tyrimo metodika

Magistro baigiamajame darbe atliekamas ieškomasis (angl. exploratory) tyrimas. Našumo vertinimo metodų pritaikymas aktorių modeliu grįstoms sistemoms nėra plačiai išnagrinėtas. Nėra suformuotų gerųjų praktikų, kurios leistų paprastai apskaičiuoti tokios sistemos našumo charakteristikas praktinėje veikloje. Dėl ieškomojo tyrimo specifikos, tyrimas vykdytas inkrementiniu būdu, tikslinant nagrinėjamą problemą, darbo tikslus. Darbo tyrimams pasirinkti ir kokybiniai, ir kiekybiniai metodai. Darbe surinkti tiek ir skaitmeniniai duomenys (gauti iš eksperimentų, skaičiavimų), tiek tekstinė informacija (iš literatūros analizės).

Gauti teoriniai rezultatai

Šiame darbe pateikiamas eilių teorijos metodo, našumo charakteristikų skaičiavimui, papildymas, leidžiantis vertinti sistemų, turinčių komunikacijos išskaidymo šabloną (*fork-join*), našumą.

Gauti praktiniai rezultatai

Praktinis šio darbo rezultatas – pasiūlytas paprastai praktiškai pritaikomas metodas aktorių modelių grįstų sistemų našumo charakteristikų apskaičiavimui.

Darbo apimtis

1 SKYRIUJE („Klasikinės sistemos“) aprašomas šiuo metu dažniausiai verslo sistemoms kurti taikoma struktūra. Įvardinami tokių sistemų trūkumai ir kylančios problemos.

2 SKYRIUJE („Reaktyvios sistemos“) pateikiama šiuo metu populiarėjančių, reaktyvių sistemų sąvoka, esminės jų savybės. Įvardinamos aktorių modelių grįstos sistemos, kurios tenkina reaktyvioms sistemoms keliamus reikalavimus.

3 SKYRIUJE („Sistemų našumo vertinimas“) aprašomas kuriamų sistemų našumo vertinimo reikalingumas, pateikiama idėja, nurodanti našumo vertinimą įtraukti į kuo ankstesnius programinės įrangos kūrimo etapus.

4 SKYRIUJE („Analizės būdai“) aprašomi populiariausi našumo vertinimo būdai – simuliacija ir analitinis metodas.

5 SKYRIUJE („Pranešimų perdavimo sistemų našumo vertinimas“) pateikiami našumo vertinimo metodai, kurie taikomi vertinant pranešimų perdavimo sistemų našumą.

6 SKYRIUJE („Našumo vertinimas naudojant eilių teoriją“) apžvelgiamas eilių teorijos metodas sistemos našumo vertinimui, pateikiami metodo keliami reikalavimai sistemai, apribojimais.

7 SKYRIUJE („Aktorių modelis“) aprašomi aktorių modelio principai ir išskiriamos pagrindinės aktorių modelio savybės.

8 SKYRIUS („Našumo vertinimo modeliai“) yra pagrindinis darbo skyrius, kuriame pateikiami atrinkti našumo vertinimo metodai, tinkami aktorių modelių grįstoms sistemoms. Pateikiami metodų aprašai, skaičiavimo pavyzdžiai. Taip pat pateikiamas eilių teorijos metodo papildymas, leidžiantis nagrinėti sistemas su išsišakojimo komunikacijos šablonu.

9 SKYRIUJE („Aktorių modelio ir našumo vertinimo modelių susiejimas“) darbe nagrinėti našumo vertinimo metodai yra susiejami su aktorių modeliu, pateikiant reikalingus apribojimus aktorių modelyje.

10 SKYRIUJE („Našumo vertinimo modelių naudojimas“) pasiūlomas praktinio našumo vertinimo metodų pritaikymas, pateikiami ir aprašomi pritaikymo pavyzdžiai.

Rezultatai ir išvados pateikiami darbo rezultatai ir išvados.

1. KLASIKINĖS SISTEMOS

Šiuo metu dažniausiai verslo sistemoms naudojama keletos lygmenų (angl. n-tier) struktūra, kurioje komunikacija vyksta tik tarp gretimų lygmenų ir tik sinchroniniu būdu. Tokios sistemos dažniausiu atveju sudaromos iš trijų lygmenų: atvaizdavimo, verslo logikos ir darbo su duomenimis. Atėjus naujai užklausiai į sistemą yra išskiriama nauja gija tos užklaustos apdorojimui ir ji yra laikoma užimta visą laiką, kol paruošiamas užklaustos rezultatas. Taip galimi atvejai, kai gija yra užsiblokavusi, tačiau joje jokie veiksmai nėra atliekami. Tokiu atveju ta gija negali būti panaudojama kitų užklaustų apdorojimui, o ekstremaliu atveju – laisvų gijų skaičius sistemoje gali baigtis, nors ir visose gijose jokie veiksmai nėra atliekami ir taip sistema nebegalės priimti naujų užklaustų. [ASF16]

Manuel *et al.* straipsnyje [MA03] pateikia keletos lygmenų architektūros trūkumus. Pagrindiniu trūkumu įvardinama tai, kad tarp atskirų lygmenų ir lygmenų viduje persiunčiamas kodas, o ne duomenys. Kodu įvardinami objektai, kurie savyje talpina ir duomenis ir funkcionalumą. Šie objektai komunikuoja tarpusavyje kviečiant vieni kitų metodus. Tokiu būdu galimi sistemos blokavimai dėl bendros būsenos (angl. shared state). Didžiausi trūkumai pastebimi išskaidytose sistemose, sudarytose iš atskirų aplikacijų. Tuomet kitų aplikacijų objektai (kodas) gali būti iš vis nepasiekiami ir juos perduoti gali nebūti galimybės. Tačiau palyginimui duomenys gali būti perduodami tarp skirtingų aplikacijų. Darbe pateikiama, kad kritinės savybės kuriant keletos lygmenų architektūros aplikacijas yra silpna sankiba (angl. Loose-coupling), stambaus granuliariškumo komunikacija, asinchroninė komunikacija. Į šias charakteristikas turi būti atkreipiamas ypatingas dėmesys ir jų įgyvendinimas keletos lygmenų architektūroje gali pareikalauti didesnių pastangų.

2. REAKTYVIOS SISTEMOS

Sistemos kurios yra reaguojančios (angl. Responsive), atsparios (angl. Resilient), elastiškos ir grįstos asinchroniniais pranešimais yra apibendrintai vadinamos reaktyviomis sistemomis. Toks apibendrinimas yra pateikiamas reaktyvių sistemų manifeste. [TRM14] Vienas iš sistemos realizavimo būdų, įgyvendinantis reaktyvių sistemų manifeste nurodytus principus, yra sistemos grįstos aktorių modeliu. Tokiose sistemose kiekvienas darbo elementas – aktorius – yra nepriklausomas, savarankiškas ir vykdo užduotis ateinančias per eilę iš kitų aktorių bei išsiunčiantis rezultatus kitiems aktoriams. Tokioje sistemoje pilnai įgyvendinama asinchroninė komunikacija, išvengiama blokavimosi, taip pat galima realizuoti nepriklausomą gijų skirstymą aktoriams – kai gija paskiriama vienam aktoriaus darbui atlikti ir nebelikus daugiau darbų ta gija yra gražinama į pulą.

Aktorių modeliu grįstose sistemose aktoriai gali sukurti kitus aktorius. Tai unikali aktorių modelio teikiama savybė, kuri leidžia pasiekti didesnę dinamiškumą, elastiškumą. Aktoriai tarpusavyje komunikuoja asinchroniniais pranešimais. Kitokių duomenų aktoriai į išorę nepateikia. Gavus naują pranešimą aktorius gali:

- siųsti baigtinį skaičių pranešimų kitiems aktoriams
- sukurti baigtinį skaičių naujų aktorių
- pakeisti elgseną, pasiruošiant kito pranešimo apdorojimui.

Grynoje aktorių sistemoje viskas yra aktoriai. Netgi ir primityvūs programavimo tipai, kaip simbolių eilutė ar sveikieji skaičiai.

Aktoriai turėtų būti mažos apimties, sudaryti svarbiausia remiantis vienos atsakomybės principo. [Ver15]

3. SISTEMŲ NAŠUMO VERTINIMAS

Ferscha darbe [Fer92] aprašomas kuriamos sistemos našumo analizės reikalingumas. Pabrėžiama ankstyvo sistemos našumo vertinimo svarba ir privalumai. Pagal tradicinį programų sistemų kūrimo procesą našumo vertinimas vykdomas visos sistemos ar atskiro etapo pabaigoje. Tai stipriai padidina rastų trūkumų taisymo kaštus. Autorius pateikia pasiūlymus, kad našumo inžinerija turi būti įtraukiama ankstyvose kuriamos sistemos projektavimo stadijose ir palaikyti sistemos kūrimo procesą visą sistemos kūrimo laiką. Tai sudarytų:

- Našumo prognozavimą ankstyvame sistemos projektavimo etape;
- Našumo analizę detalaus projektavimo ir kūrimo etapo metu;
- Našumo stebėjimą ir matavimą testavimo ir taisymo metu.

Našumo vertinimui projektavimo metu siūloma sudaryti pageidaujamo detalumo sistemos našumo specifikaciją, naudojant formalią arba iš dalies formalią kalbą. Tokią specifikaciją toliau galima panaudoti skaičiuojant našumo charakteristikas. Kadangi galima vertinti tiek atskirų sistemos dalių, tiek jų junginių našumą, todėl tokia veikla turėtų būti neužimanti daug kūrimo laiko. Siūloma sudaryti našumo šablonus, kuriuos vėliau būtų galima naudoti pakoreguoti ir greitai įvertinti pokytį. Tokį procesą autorius įvardina kaip į našumą orientuotą lygiagrečių programų sistemų kūrimą.

4. ANALIZĖS BŪDAI

Kokybinės sistemos charakteristikos, tokios kaip našumas, atsako laikas, numatomos ir įvertinamos naudojant sistemos analizę. Charakteristikų numatymui yra reikalingas tinkamo abstrakcijos lygmens sistemos modelis. Ankstyvas charakteristikų neatitikimo numatytiems reikalavimams nustatymas, naudojantis sistemos architektūros modeliu, leidžia sumažinti keitimų išlaidas.

Literatūroje pateikiama keletas kiekybinės analizės būdų: simuliacija, analitinis metodas.

4.1. Simuliaciniai metodai

Simuliaciniai metodai galimi dvejopi – į procesus orientuoti ir į įvykius orientuoti. Į procesus orientuoti metodai apibrėžia sistemą kaip tarpusavyje sąveikaujančių procesų rinkinį, kurie atkartoja realios sistemos komponentų elgseną nustatytu detalumo lygiu, priklausančiu nuo simuliacijos tikslų.

Į įvykius orientuoti simuliacijos metodai sistemą apibrėžia įvykiais, kurie iššaukia pokyčius sistemos būsenoje. Vykdamas tokį metodą, turi būti nustatytas rinkinys galimų įvykių ir kiekvienam įvykiui kokią įtaką sistemos būsenai to įvykio iššaukimas daro.

Sistemos parametrai simuliaciniam metodui yra išvedami iš sistemos architektūros aprašo. Išgavus parametrus yra sudaromas simuliacijos modelis ir po to sukuriama simuliacijos modelio realizacija specialioje simuliacijos programoje. Ši realizacija yra įvykdoma. [BM03]

Marzolla darbe pateikiamas aprašo vieną iš galimų simuliacinio modelio sudarymo metodų, sistemos našumo charakteristikoms nustatyti. Metodas remiasi UML architektūros aprašu – paima anotuotus parametrus iš diagramų ir pagal juos sudaro simuliacijos programą. Darbe teigiama, kad gaunami duomenys nėra tikslūs, tačiau tinkami pirminiam įvertinimui, turint architektūros aprašą, prieš sistemos realizaciją. Šis metodas yra pritaikytas sistemoms, kurios gauna užklausas iš išorinių vartotojų – žmonių arba kitų sistemų. Taip pat sistema turi būti sudaryta iš atskirų servisų, kuriuos užklausomis vykdys vartotojai. Simuliacijos modelis sudarytas taip, kad įvykdžius vieną servisą, vartotojas laukia nustatytą laiką, prieš kviečiant kitą servisą. Simuliacija pateikia tokius rezultatus:

- resursų naudojimą (angl. utilization)
- resursų pralaidumą (angl. throughput)
- vidutinius veiksmų ir panaudos atvejų vykdymo laikus. [Mar04]

4.2. Analitiniai metodai

Analitiniai metodai įprastu atveju yra žymiai efektyvesni ir tikslesni, nei simuliaciniai, tačiau gali būti pritaikomi tik tam tikriems, specialiai paruoštiems modeliams. Analitiniai metodai dažniausiai remiasi Petri tinklais, eilių teorija, procesų algebra. Kai kurie iš analitinių metodų skaičiavimo būdų remiasi prielaida, kad užklausos į sistemą ateina pagal statistinį skirstinį (pvz. eksponentinį). Tuomet tokie metodai visos sistemos ar jos dalies kokybines charakteristikas, tokias kaip atsako laikas, skaičiuoja remiantis statistiniais metodais. [CMI04]

Goševa–Popstojanova straipsnyje pateikia architektūra grįstų analizės būdų skirstymą:

- būseną grįsta – naudojamas valdymo eigos (angl. control-flow) grafas architektūrai nusakyti ir remiantis tuo nustatomas programinės įrangos patikimumas;
- keliais grįsta – programinės įrangos patikimumas nustatomas remiantis vykdymo keliais, kurie randami eksperimentiškai arba algoritmiškai;
- sudėtinė – laiko kad kiekvieno komponento patikimumas yra modeliuojamas Puasono procesu, tuomet komponentų klaidos taip pat modeliuojamos Puasono procesu, o visos sistemos klaidos gaunamos kaip atskirų elementų klaidų funkcijų suma. [PMT01]

5. PRANEŠIMŲ PERDAVIMO SISTEMŲ NAŠUMO VERTINIMAS

Sistemos, kuriose komunikacija tarp atskirų sistemos komponentų vyksta siunčiant pranešimus tarp jų bendrai gali būti vadinamos pranešimų siuntimo sistemomis (angl. message passing system). Aktorių modeliu grįstos sistemose atskiri aktoriai bendrauja tarpusavyje siunčiamais pranešimais, todėl tokios sistemos yra vienas iš pranešimų siuntimo sistemų tipų. Pranešimų siuntimo sistemos išsiskiria nepriklausimais, savarankiškais veikiančiais elementais ir galima asinchronine komunikacija.

Freitas ir Macedo darbe [FM13] teigiama, kad pranešimų siuntimo sistemos šiuo metu vis labiau populiarėja. Tokių sistemų našumo vertinimui siūlomas simuliacijos metodas. Straipsnyje pateikiamas simulatoriaus karkaso aprašas, kuris yra tinkamas dinamiškoms sistemoms – tokioms, kurių resursai, pasiekiamumas, apkrova kinta sistemos veikimo metu.

Aktorių modelio našumo vertinimą aprašo Jafari disertacijoje [Jaf16]. Šiame darbe autorius nagrinėja aktorių modelio realizacija Rebeca kurtų sistemų našumo vertinimą. Autorius teigia, kad „lygiagrečių išskirstytų sistemų modeliavimas ir verifikavimas reikalauja daug pastangų dėl veiklos ir struktūrinio tokių sistemų sudėtingumo“. Iškiriama, kad aktorių modelio naudojimas populiarėja. „Augant debesų kompiuterijai, web servisams ir keletos branduolių architektūroms programavimas naudojant aktorių modelį tapo vis aktualesniu“. Iškiriama, kad nors ir aktorių modeliu grįstos sistemos populiarėja, tačiau tokių sistemų analizė nėra plačiai išnagrinėta, darbų šia tema nėra daug. Sistemų našumo vertinimui išskiriami simuliacijos, modelio tikrinimo ir statistinis metodai. Konkrečiai aktorių modelio realizacijai, pavadinimu Timed Rebeca, aprašomi sukurti našumo vertinimo įrankiai. Plačiai išnagrinėjamas ir pritaikomas simuliacijos metodas. Statistinis metodas naudojamas tik įverčių nuspėjimui remiantis keletu gautų statistikos rezultatų.

Todd *et al.* darbe [TKL+11] pateikiamas Scala programavimo kalba sukurtų sistemų, grįstų aktorių modeliu našumo įvertinimas. Metodas, pasirinktas našumo įvertinimui buvo simuliacija. Buvo vykdomas sukurtos sistemos testavimas vykdant ją nustatytą skaičių kartų parenkant skirtingas sistemos apkrovas.

Vienas iš siūlomų literatūroje sistemos našumo įvertinimo būdų yra sistemos kritinio kelio paieška. Ashton ir Penny [AP91] silpnos sankibos (angl. loose coupling) išskirstytoms sistemoms siūlo sudaryti grafą, kuris atvaizduotų komunikacijos kanalus sistemoje. Tuomet pasiūlomas metodas kaip turint sistemos kritinį kelią – sistemos veikimo eigos kelią, kuris užtrunka ilgiausiai – išvesti sistemos veiklos našumo charakteristikas. Metodas yra pritaikomas konkretiems, nurodytiems atvejams ir reikalauja prielaidų tenkinimo. Kadangi autorių siūlomas

metodas yra taikomas išskirstytoms sistemoms bendrai, todėl nagrinėjant šį metodą aktorių modelio atvejui reikalinga tenkinti prielaidas, kurios netenkina aktorių modelio.

6. NAŠUMO VERTINIMAS NAUDOJANT EILIŲ TEORIJĄ

Sistemų, kuriose komunikacijoje dalyvauja eilės, našumo charakteristikų radimui yra tinkamas eilių teorijos metodas. Tai yra statistinis metodas, kuris remiasi Markovo eilėmis.

Balsamo *et al.* darbe [BMM04] pateikia metodą, kai iš turimo sistemos modelio aprašyto UML kalba galima išvesti eilių tinklą. Eilių tinklas yra komunikacijos tarp elementų su eilėmis struktūrą aprašantis grafas. Toliau turint sudarytą eilių tinklą, naudojantis eilių teorija galima rasti bazines našumo charakteristikas. Darbe pabrėžiama ankstyvos našumo analizės nauda. Tai sumažina sistemos kūrimo kaštus, leidžia palyginti galimos sistemos realizacijos alternatyvas ir identifikuoti sistemos problemines vietas. Darbe siūlomas metodas pritaikomas tik tokioms sistemoms, kurios tenkina pradinį eilių teorijos naudojimo reikalavimus.

Kitame darbe [BM05], Balsamo ir Marzolla nurodo, kad programų sistemos modelis turi būti apribotas, siekiant jį panaudoti našumo modelio sudarymui ir našumo charakteristikų radimui. Konkrečiai nurodoma, kad esant *fork-join* tipo komunikacijos šablonui sistemoje, naudoti eilių teorijos nėra galima. Siekiant apskaičiuoti tokios sistemos našumo charakteristikas reikalingas teorijos praplėtimas ir tuo uždavinys sudėtingėja. Autorius tokiu atveju siūlo naudoti kitus tinkamus metodus, pavyzdžiui simuliaciją.

7. AKTORIŲ MODELIS

Aktorių modelis yra teorinis lygiagrečios komunikacijos tarp nepriklausomų komponentų modelis. Pirmą kartą šis modelis buvo paminėtas 1973 m. Carl Hewitt ir kitų straipsnyje [HBS73]. Prieš aktorių modelį, lygiagrečios komunikacijos aprašymui buvo naudojamos žemo lygio sąvokos tokios kaip gija, užraktas, kanalas.

Aktoriumi vadinamas pagrindinis modelio komponentas, kuris turi iš anksto nustatytą elgseną ir saugo vidinę būseną.

Pagrindinė aktorių modelio savybė yra tai, kad kiekvienas aktorius sistemoje veikia nepriklausomai ir izoliuotai nuo kitų aktorių. Vienintelė komunikacijos su kitais aktoriais priemonė – asinchroninių pranešimų siuntimas. [Agh85]

Kadangi visi aktoriai sistemoje veikia nepriklausomai, todėl globalios sistemos būsenos apibrėžima yra negalimas ir nekorektiškas. Taip pat sistemoje nėra reikalingas centralizuotas visuotinis planuotojas (angl. global scheduler), kuris reguliuotų ir valdytų komunikaciją tarp atskirų aktorių.

Pranešimai aktorių modelyje siunčiami asinchroniniu būdu, išsiuntus pranešimą aktorius nelaukia atsakymo, nesiblokuoja. Atsakymas į pranešimą aktoriui modeliuojamas kaip atskiro pranešimo gavimas.

Aktoiaus išsiųstas pranešimas susideda iš:

- informacijos, persiunčiamos kitam aktoriui (pavyzdžiui, duomenų, veiksmų, kuriuos reikia atlikti, susijusių aktorių adresų);
- aktoiaus adreso, kuriam turi būti persiunčiamas rezultatas
- siuntėjo aktoiaus adresas.

Kiekvienas aktorius sistemoje identifikuojamas adresu. Aktorius gali siųsti pranešimą tik tiems aktoriams, kurių adresus žino. Sužinoti adresus galima:

- sukūrus naują aktorių;
- paminėjus aktoiaus adresą gautame pranešime.

Adreso žinomumo ryšys tarp aktorių gali būti asimetrinis, t. y. aktorius A gali žinoti aktoiaus B adresą, tačiau aktorius B, nežino aktoiaus A adreso.

Gavęs naują eilinį pranešimą aktorius gali atlikti tokius veiksmus:

- įvykdyti skaičiavimus (skriptą);
- išsiųsti pranešimus kitiems, žinomiems aktoriams;
- sukurti naujus aktorius;

- apibrėžti elgsenos pasikeitimą, kuris pradės galioti nuo sekančio pranešimo.

Dėl galimybės sukurti aktorius eigoje, aktorių modelis apibrėžia dinaminės topologijos galimybę. Aktoriaus sukurtų naujų aktorių adresai saugomi tėviniame aktoriuje ir taip galimas darbo išskaidymas.

Dėl globalios būsenos sistemoje neapibrėžtumo, nėra aiški ir sąvoka „visiems aktoriams“, todėl ir pranešimų siuntimas plataus transliavimo (angl. broadcast) būdu nėra galimas.

Aktorių modelyje yra modeliuojamas tik pranešimų gavimo įvykis, nes tik gavus pranešimą, aktorius gali atlikti su šiuo pranešimu susijusius veiksmus. Išsiųstam pranešimui aktorius įtakos padaryti negali. Taip pat, aktorių modelyje pranešimų gavimo įvykiai yra nedalomi ir pranešimų siuntimas neužima trukmės laike.

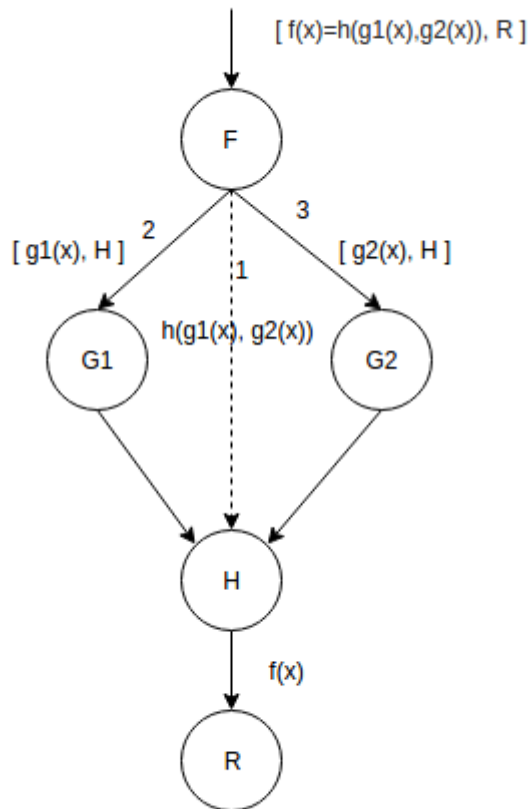
Vienu metu aktorius gali gauti ir apdoroti tik vieną pranešimą. Tam tikra skaičiavimams reikalinga informacija gali būti saugoma vidinėje aktoriaus būsenoje, tačiau ji negali būti pasiekta iš išorės, kitų aktorių. Vienintelis būsenos paskelbimo kitiems aktoriams ir būsenos keitimo būdas – pranešimų siuntimas.

Pagal aktorių modelį, nėra apibrėžiama kaip aktoriai gauna pranešimus. Įprastai tariama, kad kiekvienas aktorius turi atskirą eilę, arba, kitaip vadinamą, pašto dėžutę, į kurią pirmiausiai patenka gauti pranešimai. Tačiau tai nėra vienintelis sprendimas. Galimas ir variantas, kai eilės funkciją aktoriui atlieka kitas aktorius arba dar kitokie sprendimai.

Taip pat, nėra apibrėžta, kokia tvarka gautus pranešimus aktorius apdoros, tai paliekama spręsti kiekvienai situacijai ir realizacijoms.

Aktorių modelis neapibrėžia santykio tarp aktorių ir gijų, branduolių, procesorių. Tai taip pat paliekama spręsti konkrečioms aktorių modelio realizacijoms. [Agh86]

Toliau, 1 pav. pateikiamas *fork-join* komunikacijos modelio galimas realizavimo pavyzdys aktorių modeliu. Paveiksle apskritimais su raidėmis vaizduojami atskiri aktoriai, ištisinėmis linijomis – pranešimų siuntimo kryptys, punktyrinėmis linijomis – naujų aktorių kūrimo kryptys.



1 pav. Fork-join komunikacijos modelio realizacija aktorių modeliu.

Tarkime, reikia apskaičiuoti $f(x) = h(g_1(x), g_2(x))$. Pradinis aktorius F gauną pranešimą apskaičiuoti šį reiškinį ir nusiųsti rezultatą aktoriui R. Tuo metu aktorius F:

- sukuria naują aktorių H;
- siunčia pranešimą aktoriui G1 apskaičiuoti $g_1(x)$ ir rezultatą nusiųsti aktoriui H;
- siunčia pranešimą aktoriui G2 apskaičiuoti $g_2(x)$ ir rezultatą nusiųsti aktoriui H.

Taip garantuojama, kad aktorius H gaus abu pranešimus iš aktorių G1 ir G2, tačiau nėra apibrėžiama, kokia tvarka šie pranešimai bus gauti. Papildomai, kartu su reiškinų rezultatais, aktoriai G1 ir G2 siųs indikatorius 1 arba 2 atitinkamai, kurie rodytų kurio aktoriaus rezultatas gautas. Aktoriaus H skriptas turės sulaukti abiejų aktorių G1 ir G2 pranešimų (naudojantis gautais indikatoriais) ir gavus abu pranešimus apskaičiuoti $h(g_1(x), g_2(x))$ ir rezultatą nusiųsti aktoriui R.

Apibendrinant aktorių modelis pasižymi tokiomis pagrindinėmis savybėmis:

- Asinchroninis komunikavimas pranešimais (įvykiais);
- Aktoriaus gautų žinučių buferis (pvz.: eilės) ;

- Dinaminis aktorių elgsenos pasikeitimas;
- Dinaminė topologija (naujų aktorių sukūrimas vykdymo eigoje);
- Visuotinės būsenos nebūvimas;
- Pranešimai visuomet pasiekia tikslą;
- Gautų pranešimų apdorojimo tvarka neapibrėžta;
- Vienu metu – vienas apdorojamas pranešimas;
- Pranešimo (įvykio) siuntimas neužima laiko.

8. NAŠUMO VERTINIMO MODELIAI

Toliau darbe bus apžvelgiami trys sistemos našumo vertinimo modeliai. Visi trys modeliai yra skirti sistemoms, kurios sudarytos iš komponentų su eilėmis ir komunikacija vykdoma pranešimais tarp komponentų, pranešimui pirmiausia patenkant į eilę. Iš pradžių bus apžvelgtas eilių tinklo apibrėžimas.

8.1. Eilių tinklas

Sistema sudaryta iš eilių tinklo (angl. queueing network) yra tokia sistema, kurios pagrindinės sudedamosios dalys yra centrai su eilėmis. Centrai, sistemoje atlieka skaičiavimus. Tarp atskirų centrų yra siunčiami darbai arba pranešimai, kurie pirmiausiai patenka į kiekvieno centro eilę ir, jeigu darbo atvykimo metu centras nėra užimtas, pereina į vykdymą centre. Kiekviena eilė gali būti standartizuotai aprašyta naudojantis Kendall notaciją, kurios pilną versiją sudaro 6 parametrai $A/S/m/B/K/SD$, kur [Fra00]:

- A – trukmių tarp pranešimų į eilę atvykimo statistinis pasiskirstymas;
- S – centro darbo vykdymo laiko statistinis pasiskirstymas;
- m – serverių skaičius eilėje;
- B – maksimalus skaičius užklausų, kurį talpina eilė. Jeigu šis parametras praleidžiamas, laikoma, kad eilė yra begalinės talpos;
- K – maksimalus skaičius užklausų, kuris gali atvykti į eilę (pvz.: užklausų skaičius sistemoje). Jeigu šis parametras praleidžiamas, laikoma, kad reikšmė – begalybė.
- SD – darbų planavimo strategija (eilės tvarka, kuria vykdomi darbai iš eilės). Jeigu šis parametras praleidžiamas, laikoma, kad strategija yra FIFO (angl. First in – First out).

Dažniausios reikšmės statistinio pasiskirstymo parametrams yra:

- M – Markovo (be atminties) procesas;
- D – deterministinis pasiskirstymas (pastovus laikas, be nuokrypių);
- E_K – Erlango pasiskirstymas su parametru K ;
- G – Bendrasis pasiskirstymas (nežinomas pasiskirstymas).

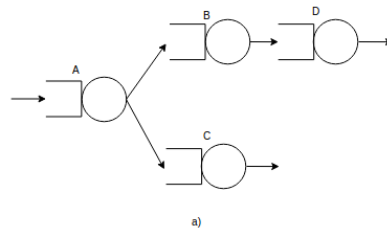
Markovo procesas pasižymi savybe, kad kiekvienas naujas įvykis yra nepriklausomas nuo anksčiau įvykusių įvykių. Atvykimo laiko modeliavimui naudojamas Puasono statistinis skirstinys. Centro darbo laiko vykdymo modeliavimui – eksponentinis skirstinys.

Šiame darbe bus nagrinėjamos tik eilės, kurios Kendall notacija aprašomos M/M/1. Tai yra tokios eilės, kurių

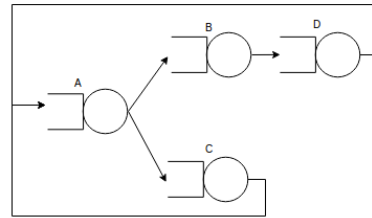
- atvykimo laikai pasiskirstę pagal Puasono skirstinį;
- centro darbo laikai pagal eksponentinį skirstinį;
- darbų atvykimo į centrą ir vykdymas centre yra nepriklausomas nuo istorijos;
- eilė turi tik vieną centrą (serverį), kuris vykdo darbus;
- eilės talpa neribojama (begalinė);
- maksimalus skaičius darbų sistemoje neribojamas;
- darbų iš eilės tvarka yra FIFO.

Dar vienas iš eilių tinklų skirstymų yra išskirstymas į atvirus eilių tinklus ir uždarus eilių tinklus [LZG+87]. Atvirame eilių tinkle darbai į sistemą patenka iš išorės ir/arba perėjus tinklą darbai yra išsiunčiami į išorę. Atviro tinklo pavyzdys pateikiamas 2 pav. a), kur rodoma, kaip darbai atėję iš išorinio šaltinio patenka į eilę A, toliau pereina į eiles B ir C, iš B eilės pereina į D eilę ir toliau iš C ir D darbai išeina į išorės tikslą (angl. sink).

Uždaruose eilių tinkluose tėra fiksuotas darbų skaičius ir jie negali pereiti į išorę atvykti naujų darbų iš išorės. Pabaigus darbą vykdyti paskutiniame centre, šis darbas persiunčiamas atgal į pirmąjį centrą. Uždaro eilių tinklo pavyzdys pateikiamas 2 pav. b), kur rodoma kaip iš eilės A darbai pereina į eiles B ir C, iš eilės B pereina į eilę D, ir tada iš eilių C ir D darbai grįžta atgal į eilę A.



a)



b)

2 pav. a) atviro eilių tinklo pavyzdys; b) uždaro eilių tinklo pavyzdys.

8.2. Tikslioji vidutinių reikšmių analizė

Uždarų eilių tinklų našumo apskaičiavimui yra naudojami iteraciniai metodai. Vienas iš metodų yra vidutinių reikšmių analizės metodas (angl. Mean Value Analysis). Šis metodas remiasi Little dėsniumi, kuris teigia, kad vidutinis klientų skaičius sistemoje yra lygus sistemos pralaidumo ir vidutinio laiko, kiek klientas praleidžia sistemoje, sandaugai, t. y.:

$$N = X \cdot R ,$$

kur N yra klientų skaičius sistemoje (vnt.), X – sistemos pralaidumas (1/sek.), R – kliento trukmė sistemoje (sek).

Vidutinių reikšmių analizės metodas susideda iš tokių 3 lygybių [Chu12]:

- Little dėsnio, išreikšto sistemos pralaidumui:

$$X(N) = \frac{N}{\sum_{i=1}^M R_i(N)} ,$$

kur $R_i(N)$ – kliento trukmė centre i, kai sistemoje yra N klientų.

- Little dėsnio, pritaikyto i-tajam centrui:

$$Q_i(N) = X(N) R_i(N) ,$$

kur $Q_i(N)$ – eilės ilgis centre i , sistemoje esant N klientų, įskaičiuojant tiek klientus esančius eilėje, tiek pačiame centre.

- Centro trukmės lygybės:

$$R_i(N) = \begin{cases} D_i & (\text{delsos centras}) \\ D_i(1 + A_i(N)) & (\text{eilės centras}) \end{cases} ,$$

kur $A_i(N)$ yra vidutinis klientų skaičius centre i , atvykus naujam klientui, D_i yra paslaugos poreikis centre i – bendras kliento praleistas laikas centre i , gali būti skaičiuojamas, naudojantis:

$$D_i = V_i \cdot S_i ,$$

kur V_i yra skaičius kartų, kiek klientas apsilanko centre i , S_i – vieno kliento apsilankymo centre i trukmė.

Delsos centru vadinamas toks centras, kuriame nėra veiklos stabdymo, o tėra tik vykdymo trukmė, t. y. būvimo centre trukmę sudaro tik vykdymo laikas, nėra stovėjimo eilėje. Eilės centras yra įprastas centras su eile, kurioje klientai laukia, prieš patekdami į centrą.

Visi lygybių parametrai, skaičiuojant sistemos našumo įverčius, būna pateikiami kaip pradiniai duomenys, išskyrus $A_i(N)$, kuris įvertinamas atskirai.

Uždarytuose tinkluose, su klientų skaičiumi N , į centrą i atvykus naujam klientui, kol jis nėra patekęs į centro eilę, vidutinis eilės ilgis yra lygus, vidutiniam eilės ilgiui, kai visame tinkle yra $N - 1$ klientų, todėl teisinga:

$$A_i(N) = Q_i(N - 1)$$

Naudojant tikslią vidutinių reikšmių analizę yra vykdomas rekursinis skaičiavimas, pradėdamas nuo eilės ilgio, kai sistemoje yra 0 klientų ir skaičiuojant didinant klientų skaičių iki reikiamo, randamas eilės ilgis esant reikiamam klientų skaičiui tinkle.

Iš to sudaromas toks tikslios vidutinių reikšmių analizės algoritmas:

```
// Inicializacija
for k = 1 → K do
    Qk = 0
end for
```



```

// Skaičiavimas
for n = 1 → N do
  sumR = 0 // trukmių centruose suma
  for i = 1 → K do
    Ri = { Di (delsos centras)
           { Di * (1 + Ai(N)) (eilės centras)
    sumR = sumR + Ri
  end for

X = n / sumR // Sistemos pralaidumas (klientų sk. / trukmė)

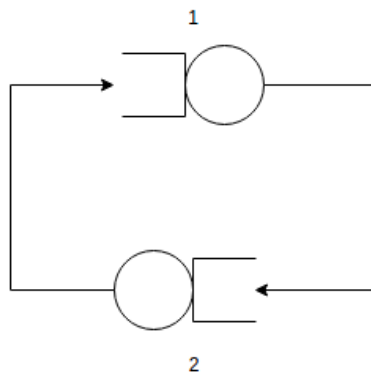
for i = 1 → K do
  Qi = X * Ri // Centro k eilės ilgis
end for
end for

```

Šiuo algoritmu gaunamas tikslus rezultatas, tačiau dėl to, kad skaičiuojama visi eilių ilgiai nuo tuščios sistemos (klientų skaičiaus sistemoje lygaus 0), iki reikiamo klientų skaičiaus, toks skaičiavimas yra neefektyvus esant didesniam klientų skaičiui [LZG+87].

8.2.1. Skaičiavimo pavyzdys

Turint 3 pav. pavaizduotą uždarą eilių tinklą ir pateiktus pradinis parametrus galima apskaičiuoti tokios sistemos našumo įverčius.



3 pav. Uždaras eilių tinklas.

Duomenys:

- Centrų skaičius sistemoje:
 $K = 2$
- Klientų skaičius sistemoje:
 $N = 3$
- Centro 1 trukmė:

$$D_1 = 2 \text{ sek}$$

- Centro 2 trukmė:

$$D_2 = 3 \text{ sek}$$

Skaičiavimas:

Eilių ilgių inicializacija:

$$Q_1 = 0$$

$$Q_2 = 0$$

Toliau klientų skaičiams n nuo 1 iki N skaičiuojama:

$$n = 1$$

$$R_1 = 2 \cdot (1 + 0) = 2$$

$$R_1 = 3 \cdot (1 + 0) = 3$$

$$\sum R = 2 + 3 = 5$$

$$X = \frac{1}{5} = 0.2$$

$$Q_1 = 0.2 \cdot 2 = 0.4$$

$$Q_2 = 0.2 \cdot 3 = 0.6$$

Toliau tęsiant skaičiavimus gaunama:

$$n = 2$$

$$R_1 = 2.8$$

$$R_2 = 4.8$$

$$X = 0.26$$

$$Q_1 = 0.728$$

$$Q_2 = 1.248$$

ir

$$n = 3$$

$$R_1 = 3.456$$

$$R_2 = 6.744$$

$$X = 0.29$$

$$Q_1 = 1.000$$

$$Q_2 = 1.956$$

Turint vidutinius eilių ilgius ir sistemos pralaidumą (X) galima apskaičiuoti ir kitus našumo rodiklius.

Sistemos atsako laikas (sek):

$$R = \frac{N}{X} = \frac{3}{0.29} = 10.345$$

Peržvelgus rezultatus, matoma, kad jie atitinka intuityviai numanomus, nes centro 2 darbo laikas yra didesnis nei centro 1, todėl darbai kaupsis centre 2, kas ir matoma rezultate (centro 2 vidutinis eilės ilgis lygus 1.956).

8.3. Apytikslė vidutinių reikšmių analizė

Siekiant pagerinti našumo skaičiavimo efektyvumą, tiksliajai vidutinių reikšmių analizei buvo pritaikyta aproksimacija, įvertinanti eilės dydis atvykus naujam klientui ($A_i(N)$).

Viena iš galimų aproksimacijų yra paskelbta Chow, kuri teigia, kad vidutinis centrų eilių ilgis esant N klientų sistemoje yra toks pat, kaip ir vidutinis centrų eilių ilgis esant vienu klientu mažiau. Ši prielaida yra teisinga, esant dideliame klientų skaičiui sistemoje.

Tarkime, apibrėžiame:

$$F_i(N) = \frac{Q_i(N)}{N} .$$

Pagal Chow prielaidą, išvedame, kad:

$$F_i(N) \approx F_i(N-1)$$

$$\frac{Q_i(N)}{N} \approx \frac{Q_i(N-1)}{N-1}$$

$$Q_i(N-1) \approx \frac{N-1}{N} \cdot Q_i(N) .$$

Remiantis tiksliaja vidutinių reikšmių analize:

$$A_i(N) = Q_i(N-1) ,$$

todėl:

$$A_i(N) \approx \frac{N-1}{N} \cdot Q_i(N) .$$

Pagal aproksimaciją pritaikomas ir vidutinių reikšmių analizės algoritmas [Chu12]:

1. Inicializuojama $Q_i(N) = \frac{N}{K}$ visiems centrams i
2. Apskaičiuojama aproksimacija: $A_i(N) \approx \frac{N-1}{N} \cdot Q_i(N)$
3. Suskaičiuojami nauji eilių ilgiai $Q_i(N)$, naudojantis lygybėmis:

$$R_i(N) = \begin{cases} D_i & (\text{delsos centras}) \\ D_i(1+A_i(N)) & (\text{eilės centras}) \end{cases}$$

$$X(N) = \frac{N}{\sum_{i=1}^M R_i(N)}$$

$$Q_i(N) = X(N) R_i(N)$$

4. Jeigu 3 žingsnyje gautas $Q_i(N)$, lyginant su naudotu 2 žingsnyje netenkina iš anksto nustatytos tolerancijos ribos, grįžtama į 2 žingsnį, kitu algoritmas sustabdomas.

8.3.1. Skaičiavimo pavyzdys

Turint tokį pat eilių tinklą, koks buvo naudotas tikslios vidutinių reikšmių analizės atveju (3 pav.) ir priimant tokius pačius pradinius duomenis skaičiuojama:

Duomenys:

- Centrų skaičius sistemoje:
 $K = 2$
- Klientų skaičius sistemoje:
 $N = 3$
- Centro 1 trukmė:
 $D_1 = 2 \text{ sek}$
- Centro 2 trukmė:
 $D_2 = 3 \text{ sek}$
- Tolerancija:
 $\Delta_i < 0.1$

Skaičiavimas:

Inicializacija:

$$Q_1 = Q_2 = \frac{N}{K} = 1.5$$

1 iteracija

$$A_1(N) = \frac{2}{3} \cdot 1.5 = 1$$

$$A_2(N) = \frac{2}{3} \cdot 1.5 = 1$$

$$R_1 = 2 \cdot (1+1) = 4$$

$$R_2 = 3 \cdot (1+1) = 6$$

$$X = \frac{3}{10} = 0.3$$

$$Q_1 = 0.3 \cdot 4 = 1.2$$

$$Q_2 = 0.3 \cdot 6 = 1.8$$

$$\Delta_1 = |1.5 - 1.2| = 0.3$$

$$\Delta_2 = |1.5 - 1.8| = 0.3$$

2 iteracija

$$R_1 = 3.6$$

$$R_2 = 6.6$$

$$X = 0.29$$

$$Q_1 = 1.044$$

$$Q_2 = 1.914$$

$$\Delta_1 = |1.2 - 1.044| = 0.156$$

$$\Delta_2 = |1.8 - 1.914| = 0.114$$

3 iteracija

$$R_1 = 3.392$$

$$R_2 = 6.828$$

$$X = 0.294$$

$$Q_1 = 0.997$$

$$Q_2 = 2.007$$

$$\Delta_1 = |1.044 - 0.997| = 0.047$$

$$\Delta_2 = |1.914 - 2.007| = 0.093$$

Po 3 iteracijos tolerancija yra tenkinama, todėl skaičiavimas sustabdomas.

Kaip galima pastebėti, rezultatai buvo gauti panašūs su tam tikra paklaida į tikslaus skaičiavimo rezultatus.

Šis metodas yra tinkamesnis esant didesniai nagrinėjamam klientų skaičiui.

8.4. Eilių teorija

Vienas iš metodų atviro eilių tinklo našumo skaičiavimui yra naudojantis eilių teorija. Eilių teorija remiasi įeinančio ir išėinančio klientų srauto lyginimu. Eilių teorijos pritaikymui yra reikalingos tokios prielaidos:

- laikas tarp klientų atvykimo į sistemą tenkina Puasono skirstinį su klientų atvykimo dažniu λ ;
- centrų darbo trukmės laikai tenkina eksponentinį skirstinį, su centro pralaidumu μ_i ;
- vienu metu gali įvykti tik vienas įvykis pakeičiantis būseną sistemoje.

Pagal eilių teoriją, eilių tinklo, susidedančio iš M eilių, būseną apibūdinama kaip vektorius $\underline{n} = (n_1, n_2, \dots, n_M)$, kur n_i yra klientų skaičius i -tojoje eilėje.

Tikimybės srautas (angl. probability flux) – vidutinis skaičius kartų per sekundę, kiek įvyksta pokyčio įvykis. Pavyzdžiui, jeigu pokytis turi dažnį 10 kartų per sekundę ir tikimybė, pagal kurią pokytis įvyksta yra lygi 0.1, tai pokytis įvyksta vieną kartą per sekundę.

Tikimybės srautas yra lygus būsenos tikimybės ir dažnio sandaugai.

Pagal eilių teoriją tikimybės srautas, pagal kurią patenkama į būseną yra lygus tikimybės srautui, pagal kurią būseną paliekama.

Toliau 1_i žymimas vektorius, kurio visos komponentos lygios 0, išskyrus i -ąją lygią 1 ir $\underline{n} - 1_i$ žymi vektorių, kurio i -ajame elemente yra $n-1$.

r_{ij} Žymimas požymis (0 arba 1), kad klientai iš eilės i yra persiunčiami į eilę j (eilė i ir j sujungtos ryšiu).

Būseną \underline{n} paliekama, kai:

1. Į sistemą ateina naujas klientas iš šaltinio. Šio atvejo tikimybės srautas yra lygus $\lambda \cdot p(\underline{n})$
2. Klientas, esantis centre i , pereina į kitą eilę arba į sistemos išorę (angl. sink). Tikimybės

$$\text{srautas: } \sum_{i=1}^M \mu_i \cdot p(\underline{n})$$

Į būseną \underline{n} patenkama, kai:

1. Klientas iš išorės šaltinio ateina į eilę i , kai joje yra $n_i - 1$ klientas. Tikimybės srautas:

$$\sum_{i=1}^M \lambda r_{si} p(\underline{n} - 1_i)$$

2. Klientas iš eilės i , pereina į išorę, kai eilėje i yra $n + 1$ klientų. Tikimybės srautas

$$\sum_{i=1}^M \mu_i r_{id} p(n+1_i)$$

3. Klientui išvykus iš eilės i , į eilę j , kai pirmojoje eilėje buvo $n + 1$ klientas, o antrojoje $n -$

$$1 \text{ klientas. Tikimybės srautas } \sum_{i=1}^M \sum_{j=1}^M \mu_j r_{ji} p(n+1_j - 1_i) .$$

Gaunama globalaus sistemos balanso lygtis:

$$\lambda p(n) + \sum_{i=1}^M \mu_i p(n) = \sum_{i=1}^M \lambda r_{si} p(n-1_i) + \sum_{i=1}^M \mu_i r_{id} p(n+1_i) + \sum_{i=1}^M \sum_{j=1}^M \mu_j r_{ji} p(n+1_j - 1_i)$$

Toliau, pagal eilių teoriją apibrėžiamos srauto lygybės, kurios nurodo kiekvienos eilės pralaidumą (θ_i). Srauto lygybės forma yra:

$$\theta_i = r_{si} \lambda + \sum_{j=1}^M r_{ji} \theta_j \quad i=1,2,3,\dots,M$$

Ji nurodo, kad eilės i pralaidumas yra lygus klientų srautui ateinančiam iš išorės bei srautų, ateinančių iš kitų eilių sumai.

Nagrinėjant kiekvieną būseną atskirai, o ne visos sistemos būseną, galima išvesti ir lokalaus balanso lygtį:

$$\theta_i p(n-1_i) = \mu_i p(n)$$

Atliekant globalaus balanso lygties pertvarkymus, naudojantis lokalaus balanso lygtimi ir srauto lygybe, gaunamas globalaus balanso lygties sprendinys:

$$\lambda = \sum_{i=1}^M r_{id} \theta_i$$

Ši lygybė sulygina klientų atvykimo ir išvykimo iš sistemos srautus ir, pagal iškeltas prielaidas, yra akivaizdžiai teisinga. Iš to išvedama, kad naudota lokalaus balanso lygtis taip pat teisinga. Pertvarkius lokalaus balanso lygtį gaunamas reiškinys:

$$p(\underline{n}) = \frac{\theta_i}{\mu_i} p(\underline{n} - 1_i)$$

Rekursiškai skaičiuojant šį reiškinį prieinama prie atviro eilių tinklo sprendinio:

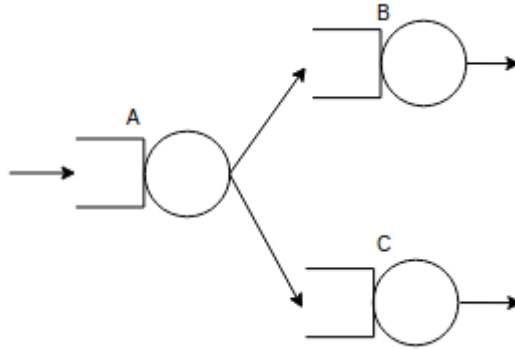
$$p(\underline{n}) = p(\underline{0}) \prod_{i=1}^M \left(\frac{\theta_i}{\mu_i} \right)^{n_i}$$

Naudojant šį sprendinį galima rasti nustatyto klientų skaičiaus sistemoje tikimybes [Rob00].

8.4.1. Atviras eilių tinklas su išsišakojimais

Ankstesniame skyriuje aprašytas atviro eilių tinklo sprendinys yra apribotas tik eilių tinklams, kuriuose klientai tarp centrų eina tiesiškai, t. y. nėra galima klientui iš centro pereiti į kitus 2 centrus. Pavyzdžiui, 4 pav. parodytame eilių tinkle, klientui išeinant iš centro A, galima pereiti tik į eilę B, arba į eilę C, pagal tam tikras nustatytas tikimybes. Tačiau nėra galimybės klientui iš eilės A pereiti vienu žingsniu ir į eilę B, ir į eilę C.

Toks komunikacijos būdas atitinka *fork-join* tipo komunikacijos šabloną sistemose, kuris yra populiarus realizuojant daugumą išskirstytų, iš atskirų nepriklausomų elementų susidedančių sistemų.



4 pav. Atviro eilių tinklo su išsišakojimu pavyzdys

Toks išsišakojimas, klientui iš vienos eilės pereinant į dvi globalaus balanso lygtyje pridėtų papildomą būsenos pasikeitimą, papildomą atvejį, kaip patenkama į būseną \underline{n} :

- Klientui išvykus iš eilės i , išsišakojant į eiles j ir k , kai eilėje i yra $n + 1$ klientas, j ir k

eilėse – po $n - 1$ klientą. Tikimybės srautas: $\sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M \mu_i r_{ijk} p(\underline{n}+1_i-1_j-1_k)$, kur

$$r_{ijk} = r_{ij} \cdot r_{ik} \cdot$$

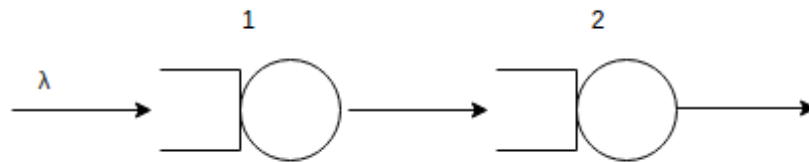
Taigi, pradinė globalaus balanso lygtis atrodytų taip:

$$\lambda P(n) + \sum_{i=1}^M \mu_i P(n) = \sum_{i=1}^M \lambda r_{si} p(n-1_i) + \sum_{i=1}^M \mu_i r_{id} p(n+1_i) + \sum_{i=1}^M \sum_{j=1}^M \mu_j r_{ji} p(n+1_j-1_i) + \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M \mu_i r_{ijk} p(n+1_i-1_j-1_k)$$

Parodžius, kad iš šios lygties išvedama akivaizdžiai teisinga lygybė, naudojantis lokalaus balanso lygtimi, būtų įrodoma, kad atviro eilių tinklo sprendinį galima naudoti ir eilėms su aprašytu išsišakojimu, o tai reiškia, kad metodas yra tinkamas sistemoms, su *fork-join* tipo komunikacijos šablonu, nagrinėti. Šiame darbe toks išvedimas nėra pateikiamas.

8.4.2. Skaičiavimo pavyzdys

Toliau pateikiamas atviro eilių tinklo našumo reikšmių apskaičiavimo pavyzdys. Turint eilių tinklą, kuris parodytas 5 pav. ir priimant pateiktus pradinis duomenis, skaičiuojama:



5 pav. Atviro eilių tinklo pavyzdys

Duomenys:

Darbai atvyksta pagal Puasono skirstinį, kur λ vidutinis darbų atvykimo dažnis (1 / sek)

Centruose darbai vykdomi pagal eksponentinį skirstinį, kur μ vidutinis centro darbų įvykdymo dažnis (1/sek)

$$\lambda = 2, \mu_1 = 3, \mu_2 = 4$$

$$r_{s1} = 1, r_{s2} = 0, r_{11} = 0, r_{12} = 1, r_{21} = 0, r_{22} = 0$$

Srauto lygtys:

$$\theta_1 = r_{s1} \lambda + (r_{11} \theta_1 + r_{21} \theta_2)$$

$$\theta_2 = r_{s2} \lambda + (r_{12} \theta_1 + r_{22} \theta_2)$$

$$\theta_1 = \lambda$$

$$\theta_2 = \theta_1$$

$$\theta_1 = \theta_2 = \lambda$$

Tikimybės, kad eilėje yra n_i darbų:

$$p_1(n_1) = \left(1 - \frac{\lambda}{\mu_1}\right) \left(\frac{\lambda}{\mu_1}\right)^{n_1}$$

$$p_2(n_2) = \left(1 - \frac{\lambda}{\mu_2}\right) \left(\frac{\lambda}{\mu_2}\right)^{n_2}$$

$$p_1(n_1) = \left(1 - \frac{2}{3}\right) \left(\frac{2}{3}\right)^{n_1} = \frac{1}{3} \left(\frac{2}{3}\right)^{n_1}$$

$$p_2(n_2) = \left(1 - \frac{2}{4}\right) \left(\frac{2}{4}\right)^{n_2} = \frac{1}{2} \left(\frac{1}{2}\right)^{n_2} = \left(\frac{1}{2}\right)^{n_2+1}$$

Tikimybės, kad eilė nėra tuščia:

$$\rho_1 = \frac{\theta_1}{\mu_1} = \frac{\lambda}{\mu_1} = \frac{2}{3} = 0.667$$

$$\rho_2 = \frac{\theta_2}{\mu_2} = \frac{\lambda}{\mu_2} = \frac{2}{4} = 0.5$$

Vidutinis darbų skaičius eilėse:

$$E(L_1) = \frac{\rho_1}{1 - \rho_1} = \frac{0.667}{0.333} = 2$$

$$E(L_2) = \frac{\rho_2}{1 - \rho_2} = \frac{0.5}{0.5} = 1$$

Centrų atsako laikai:

$$E(W_1) = \frac{1/\mu_1}{1 - \rho_1} = \frac{1/3}{1 - 0.667} = 1$$

$$E(W_2) = \frac{1/\mu_2}{1 - \rho_2} = \frac{1/4}{1 - 0.5} = 0.5$$

Visos sistemos atsako laikas:

$$E(W) = \frac{1}{\lambda} (E(L_1) + E(L_2)) = \frac{1}{2} (1 + 0.5) = 0.75$$

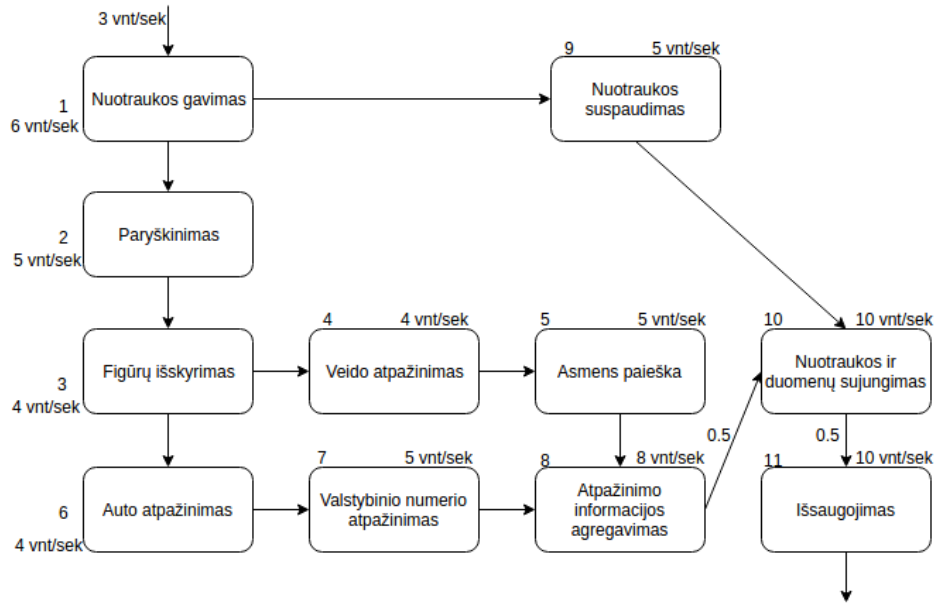
8.4.3. Skaičiavimo tikrinimas simuliacija

Darbe pateikto eilių teorijos papildymo pasiūlymo patikrinimui buvo naudojamas Java Modelling Tools (JMT) simulatorius JSIMgraph. Šis simulatorius yra skirtas komunikacijos eilių tinkluose ir Petri tinkluose simuliacijai. Įrankis suteikia grafinį vartotojo interfeisą, kuriuo naudojantis yra sudaromas nagrinėjamas eilių tinklas. Toliau nustatomi įeigos pranešimų srauto parametrai ir kiekvienos elemento (eilės) pralaidumo parametrai, elementų tarpusavio jungimosi taisyklės. [SBC17]

Naudojamas simulatorius realizuotas Java programavimo kalba. Realizacija remiasi pranešimų siuntimo komunikacija. Atskiri simuliaciją sudarantys elementai paskirstomi į atskiras vykdymo gijas. Pagrindinis JSIMgraph simulatoriaus modulis yra diskrečių įvykių simuliacijos variklis, kuris veikia kaip pranešimų siuntėjas. Simuliuojant kiekvienas įvykis, toks kaip naujo darbo atvykimas arba darbo išvykimas iš centro atitinka pranešimus su specifiniu identifikaciniu kodu. Kiekviena eilė sudaryta iš trijų dalių: pranešimų eilės, serverio, maršrutizatoriaus.

Simuliacija sustabdoma, kai našumo charakteristikos gali būti įvertinamos pakankamai tiksliai su nurodyta paklaida.

Eilių teorijos papildymo pasiūlymo teisingumo patikrinimui skaičiuojami 6 pav. parodytos sistemos našumo charakteristikų įverčiai eilių teorijos būdu ir simuliacija.



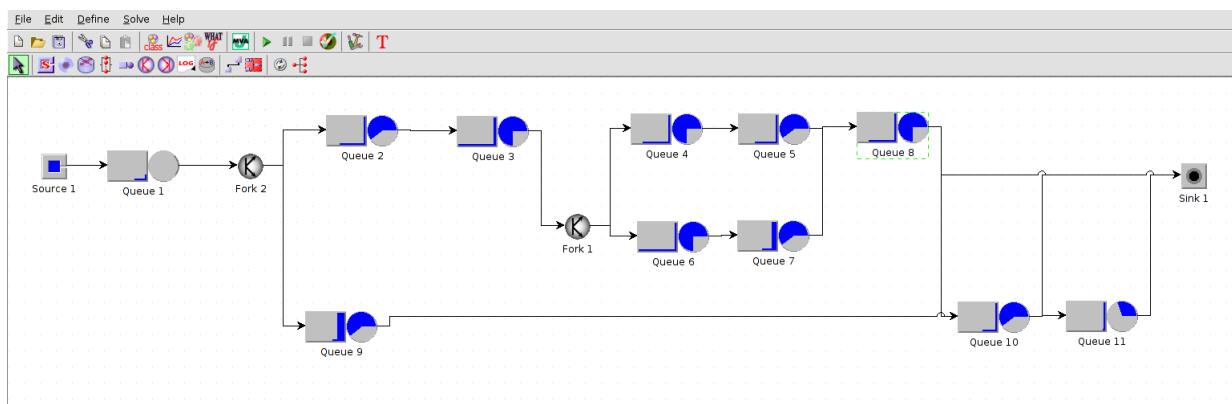
6 pav. Sistemos pavyzdys rezultatų tikrinimui vykdant simuliaciją

Pirmiausia našumo charakteristikos buvo apskaičiuotos naudojantis eilių teorijos formulėmis. Buvo tariama, kad pradinis srautas į sistemą ateinantis iš išorės yra 3 vnt/sec. Taip pat buvo duotas kiekvieno aktoriaus sistemoje pralaidumas (vnt/sec). Gauti rezultatai pateikti 1 lentelėje. Bendras visos sistemos vidutinis atsako laikas apskaičiuotas lygiu $E(W) = 6,98$ sek.

1 lentelė. Našumo charakteristikų skaičiavimo rezultatai, skaičiuojant naudojantis eilių teorija

Eilės nr. (i)	Pralaidmas (μ , vnt/sec.)	Srautas (θ , vnt/sec.)	Išnaudojamas (ρ)	Vid. Eilės ilgis ($E(L_i)$, vnt)	Vid. Atsako laikas ($E(W_i)$, sek)
1	6	3	0,5	1	0,33
2	5	3	0,6	1,5	0,5
3	4	3	0,75	3	1
4	4	3	0,75	3	1
5	5	3	0,6	1,5	0,5
6	4	3	0,75	3	1
7	5	3	0,6	1,5	0,5
8	8	6	0,75	3	0,5
9	5	3	0,6	1,5	0,5
10	10	6	0,6	1,5	0,25
11	10	3	0,3	0,43	0,14

Toliau buvo sudaromas simuliacijos modelis JMT įrankyje. Sudaryto simuliacijos modelio pavyzdys pateikiamas 7 pav.



7 pav. Simuliacijos modelio pavyzdys sudarytas JMT įrankyje

Atlikus simuliacija, gauti našumo charakteristikų įverčiai pateikti 2 lentelėje.

2 lentelė. Atlikus sistemos simuliaciją gautų našumo charakteristikų įverčiai

Eilės nr. (i)	Pralaidmas (μ , vnt/sek.)	Srautas (θ , vnt/sek.)	Išnaudojamas (ρ)	Vid. Eilės ilgis ($E(L_i)$, vnt)	Vid. Atsako laikas ($E(W_i)$, sek)
1	6	2,980	0,499	1,003	0,333
2	5	2,994	0,592	1,512	0,500
3	4	2,979	0,746	3,033	0,990
4	4	2,978	0,740	3,050	0,996
5	5	2,995	0,592	1,511	0,502
6	4	2,981	0,739	3,061	1,008
7	5	2,982	0,596	1,515	0,502
8	8	5,936	0,740	3,044	0,568
9	5	2,978	0,591	1,518	0,506
10	10	5,954	0,595	1,546	0,258
11	10	2,966	0,297	0,426	0,143

Palyginus eilių teorija apskaičiuotas našumo charakteristikas ir gautas vykdant simuliaciją, matoma, kad pirmojo ir antrojo būdo rezultatų skirtumai patenka į tolerancijos ribą lygią 0,1. Remiantis tuo galima teigti, kad charakteristikų įverčiai, gaunami naudojantis eilių teorija su

darbe pasiūlytu papildymu yra teisingi. Simuliacijos metodo naudojimą našumo charakteristikų įvertinimui riboja nagrinėjamos sistemos elementų skaičius. Didesnis atskirų elementų kiekis reikalauja aukštesnių kompiuterio, kuriame vykdoma simuliacija, resursų. Taip pat dėl simulatoriaus pagrindo realizacijos (atsitiktinių skaičių generatoriaus, patikimumo vertinimo ir nuokrypių vertinimo) mažėja galimas rezultatų tikslumas.

9. AKTORIŲ MODELIO IR NAŠUMO VERTINIMO MODELIŲ SUSIEJIMAS

Siekiant atlikti aktorių modeliu grįstos sistemos našumo vertinimą galima pasinaudoti eilių tinklų našumo vertinimo modeliais. Tačiau, naudojantis šiais modeliais iškyla apribojimai nagrinėjamam aktorių modeliui:

- aktorių topologija nesikeičia sistemos veikimo metu;
- aktoriaus elgsena nesikeičia sistemos veikimo metu;
- gauti pranešimai apdorojami FIFO tvarka;
- pranešimai į sistemą atvyksta atsitiktinai pagal Puasono skirstinį;
- aktorių veikimo trukmė atsitiktinė, pagal eksponentinį skirstinį.

Eilių tinklų sąvokų ir aktorių modelio sąvokų susiejimas galimas toks:

Eilių tinkas	Aktorių modelis
Klientas	Pranešimas
Centras	Aktorius
Eilė	Eilė (pašto dėžutė)

Našumo vertinimo metodai turėtų būti parenkami pagal sistemos struktūrą ir nagrinėjamą klientų skaičių. Esant uždarei aktorių modelio sistemai, kai yra pradinis pranešimų rinkinys, kuris vaikšto tarp aktorių ir nauji pranešimai neateina iš išorės bei esami neišeina į išorę, tuomet gali būti renkamas vidutinių reikšmių analizės metodas, priklausomai nuo klientų skaičiaus – tikslio arba apytikslės.

Jeigu aktorių modelio sistema atvira, t. y. pranešimai į sistemą ateina iš išorės, tokios sistemos našumo vertinimui tinkamas eilių teorijos modelis.

10. NAŠUMO VERTINIMO MODELIŲ NAUDOJIMAS

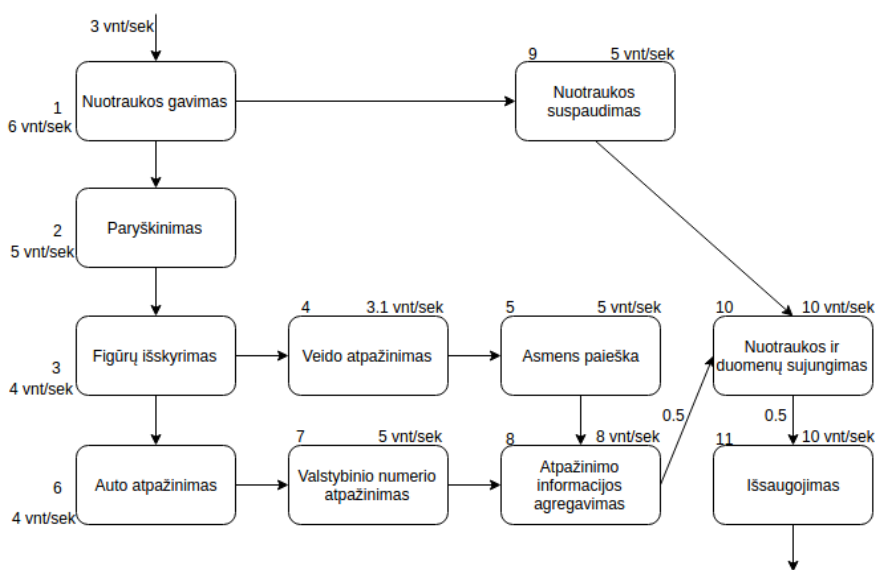
Galima išskirti keletą našumo vertinimo modelių pritaikymo, nagrinėjant ir dirbant su aktorių modelių grįstomis programų sistemomis, atvejų.

Vienas iš panaudojimo būdų – sistemos struktūros pakeitimų įtakos vertinimui. Našumo modelių naudojimas leidžia įvertinti įtaką atlikus tam tikrus aktorių struktūros pakeitimus, tokius kaip vieno aktoiaus išskaidymą į keletą atskirų. Tokį įvertinimą galima atlikti dar prieš realų pakeitimo įgyvendinimą, turint tik sistemos modelį, kas leidžia sumažinti ar išvengti galimus nuostolius.

Kitas panaudojimo atvejis – minimalaus žemesnio lygio resursų, pavyzdžiui, gijų, reikalingų sistemos veikimui, skaičiaus nustatymas. Kadangi iš našumo vertinimo metodų rezultate gaunami eilių ilgiai ir vidutinės vykdymo trukmės, iš šios informacijos galima išvesti koks reikalingas gijų kiekis, kad sistema veiktų efektyviai. Lygiai taip pat galima nustatyti ir maksimalų skaičių gijų, kurias išnaudoti sistema pajėgs vykdydama numatytą veiklą.

10.1. Sistemos struktūros pakeitimų įtakos vertinimas

Šiame skyriuje bus pateiktas pokyčių sistemos struktūroje įtakos įvertinimo pavyzdys. Pavyzdžiui turima automobilių greičio fiksavimo kamerų duomenų apdorojimo sistema, kuri iš nuotraukos atpažįsta automobilio duomenis, juos apdoroja ir išsaugo. Tokios sistemos preliminarus pavyzdys, realizuotas aktorių modelių pateiktas 8 pav. Pateiktoje schemeje



8 pav. Greičio kamerų apdorojimo sistemos, realizuotos aktorių modelių, struktūra

stačiakampiais su apvalintais kampais vaizduojami atskiri aktoriai. Kiekvienam aktoriui nurodytas jo pralaidumas – pranešimais per sekundę. Į sistemą pranešimai ateina iš išorės dažniu 3 pranešimai per sekundę. Rodyklės nurodo kryptį, kuriomis juda pranešimai tarp aktorių. Ant rodyklių nurodyti skaičiai nurodo tikimybes, kad pranešimas bus išsiunčiamas. Pavyzdžiui aktorius su numeriu 10 gauna pranešimus iš aktorių 8 ir 9, o aktoriui 11 pranešimą siunčia su tikimybe 0,5.

Tariama, kad sistema tenkina tokias prielaidas:

- Pranešimai į sistemą atvyksta atsitiktinai pagal Puasono skirstinį su nurodytu dažniu
- Aktorių darbo trukmė atsitiktinė, pasiskirsčiusi pagal eksponentinį skirstinį su nurodytu dažniu.

Naudojantis darbe nurodytais metodais apskaičiuojamos tokios aktorių sistemos našumo charakteristikos. Skaičiavimo rezultatai pateikiami 3 lentelėje. Lentelėje žymima:

- i - aktoriaus eilės numeris,
- μ – aktoriaus pralaidumas,
- θ – pro aktorių praeinančių pranešimų srautas (vnt/sek),
- $E(L_i)$ – vidutinis eilės ilgis ties kiekvienu aktoriumi, įskaitant ir pranešimą, kuris yra apdorojamas aktoriuje (vnt),
- $E(W_i)$ – vidutinis aktoriaus atsako laikas (sek).

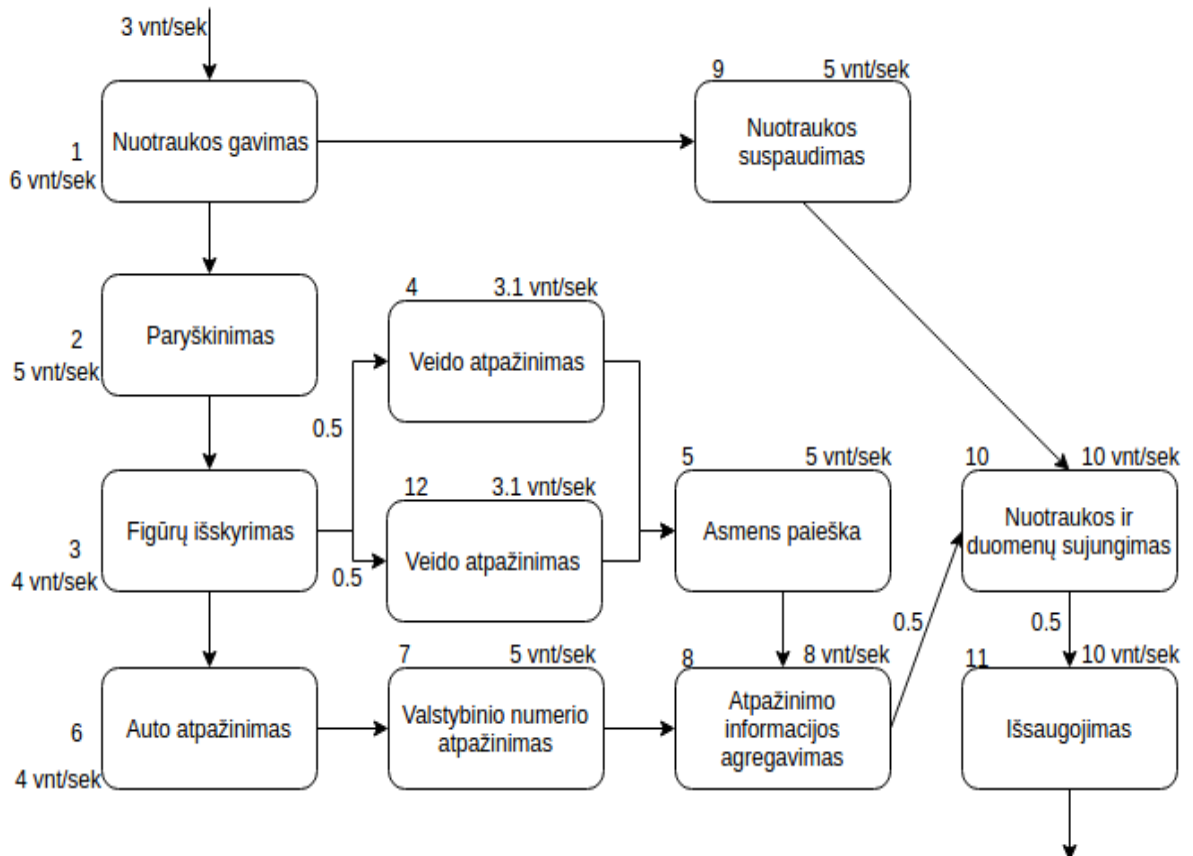
3 lentelė. Pradinės sistemos našumo charakteristikos.

Eilės nr. (i)	Pralaidmas (μ , vnt/sek.)	Srautas (θ , vnt/sek.)	Išnaudojamas (ρ)	Vid. Eilės ilgis ($E(L_i)$, vnt)	Vid. Atsako laikas ($E(W_i)$, sek)
1	6	3	0,5	1	0,33
2	5	3	0,6	1,5	0,5
3	4	3	0,75	3	1
4	3,1	3	0,97	30	10
5	5	3	0,6	1,5	0,5
6	4	3	0,75	3	1
7	5	3	0,6	1,5	0,5
8	8	6	0,75	3	0,5
9	5	3	0,6	1,5	0,5
10	10	6	0,6	1,5	0,25
11	10	3	0,3	0,43	0,14

Apskaičiavus bendrą pradinės sistemos atsako laiką gaunama (sek):

$$E(W) = 15,98$$

Toliau, pavyzdžiui, atsiradus poreikiui aktorius numeris 4 išskaidomas į du atskirus, kurie vykdys tą patį darbą. Taigi aktorius 3 pranešimus siųs su tikimybe 0,5 į 4 aktorių ir su tikimybe 0,5 siųs pranešimus naujam, 12 aktoriumi. Naujos aktorių sistemos schema pateikiama 9 pav.



9 pav. Pakeistos greičio kamerų apdorojimo aktorių sistemos struktūra

Atliekant naujos struktūros našumo charakteristikų skaičiavimus, gaunami nauji rezultatai. Šie rezultatai pateikiami 4 lentelėje. Reikšmių žymėjimai lentelėje tokie patys, kaip aprašyta aukščiau. Apskaičiavus bendrą sistemos atsako laiką gaunama (sek):

$$E(W) = 6,60$$

Kaip matoma, toks pakeitimas pagerina bendrą sistemos atsako laiką. Taip pat atskirų aktorių vidutiniai eilių ilgiai ir atsako laikai taip pat nepadidėjo. Taigi, toks pakeitimas tinkamas sistemai, pagerina jos savybes.

Taip praktiniuose taikymuose galėtų būti vykdomas aktorių modeliu grįstos sistemos struktūros pokyčio įtakos įvertinimas.

4 lentelē. Pakeistos sistemos našumo charakteristikos.

Eilēs nr. (i)	Pralaidmas (μ, vnt/sek.)	Srautas (θ, vnt/sek.)	Išnaudojamumas (ρ)	Vid. Eilēs ilgis ($E(L_i)$, vnt)	Vid. Atsako laiks ($E(W_i)$, sek)
1	6	3	0,5	1	0,33
2	5	3	0,6	1,5	0,5
3	4	3	0,75	3	1
4	3,1	1,5	0,48	0,94	0,63
5	5	3	0,6	1,5	0,5
6	4	3	0,75	3	1
7	5	3	0,6	1,5	0,5
8	8	6	0,75	3	0,5
9	5	3	0,6	1,5	0,5
10	10	6	0,6	1,5	0,25
11	10	3	0,3	0,43	0,14
12	3,1	1,5	0,48	0,94	0,63

REZULTATAI IR IŠVADOS

Šiame darbe buvo apžvelgtas aktorių modelis ir našumo vertinimo metodai, kurie tinkami aktorių modelio nagrinėjimui. Pastebėtas trūkumas eilių teorijoje, naudojant ją aktorių sistemos našumo vertinimui – komunikacijos išskaidymo (*fork-join*) šablono sistemose nepalaikymas. Darbe buvo pateiktas pasiūlymas, kaip papildyti metodą įgalinant nagrinėti ir tokius komunikacijos šablonus. Pateikto pasiūlymo teisingumas buvo patikrintas pateikiant trivialių pavyzdžių skaičiavimą bei atliekant tikrinimą palyginant rezultatus gautus skaičiuojant naudojantis eilių teorija ir naudojantis sistemos komunikacijos modelio simuliacija.

Taip pat pateiktas metodo naudojimo ir pritaikymo praktikoje pavyzdys. Darbe parodyta, kad siūlomas sistemos našumo vertinimo metodas yra paprastai praktiškai pritaikomas, kadangi jo skaičiavimus sudaro tiesinių lygčių sistemos išsprendimas ir našumo įverčių apskaičiavimas pagal formules.

Darbo rezultatas:

Pasiūlytas našumo charakteristikų skaičiavimo metodas, kuris pritaikomas praktiškai, naudojant paprastas priemones.

Atlikus darbą buvo prieita prie išvados:

Pasiūlytas našumo charakteristikų skaičiavimo metodas yra praktiškai pritaikomas ir paprastas naudoti nepriklausomai nuo sistemos dydžio.

Eilių teorijos metodai yra lengvai praktiškai pritaikomi aktorių modeliu grįstų sistemų našumo charakteristikų skaičiavimui.

ŠALTINIAI

- [Agh85] G. Agha. Semantic considerations in the actor paradigm of concurrent computation, 1985, Springer, Berlin, Heidelberg.
- [Agh86] G. Agha. Actors: a model of concurrent computation in distributed systems, 1986, MIT Press Cambridge, MA, USA.
- [AP91] P. Ashton, J. Penny, The interaction network: a performance evaluation tool for loosely-coupled distributed systems. University of Canterbury, 1991.
- [ASF16] The Apache Software Foundation. Apache Tomcat 7, 2016. [Žiūrėta 2016-06-10]. Prieiga per internetą: <<https://tomcat.apache.org/tomcat-7.0-doc/config/http.html>>.
- [BM03] S. Balsamo, M. Marzolla. A Simulation-based Approach to Software Performance Modeling, SIGSOFT Softw. Eng. Notes, 28(5), 2003, pp. 363-366.
- [BM05] S. Balsamo, M. Marzolla. Performance Evaluation of UML Software Architectures with Multiclass Queueing Network Models. Proceedings of the 5th International Workshop on Software and Performance, 2005, pp. 37-42.
- [BMM04] S. Balsamo, R. Mamprin, M. Marzolla, Performance Evaluation of Software Architectures with Queueing Network Models. Proceedings of the 1998 Workshop on Software and Performance, ACM Press, 2004, pp. 178-190.
- [Chu12] A. Chugh. Algorithms for System Performance Analysis. MEng thesis, Imperial College, London.
- [CMI04] V. Cortellessa, A. Di Marco, P. Inverardi. Model-Based Software Performance Analysis, Springer-Verlag Berlin Heidelberg, 2004, 190 pages.
- [Fer92] A. Ferscha. A Petri Net Approach for Performance Oriented Parallel Program Design, Journal of Parallel and Distributed Computing 15(3), 1992. pp. 188-206.

- [FM13] A. E. Silva Freitas, R. J. De Arajo Macedo. Performance Evaluation in Hybrid and Dynamic Distributed Systems, SIGOPS Oper. Syst. Rev. 48(1), 2014, pp. 11-18.
- [Fra00] R. F. Franks. Performance analysis of distributed server systems, 2000 Carleton University Ottawa.
- [HBS73] C. Hewitt, P. Bishop, R. Steiger. A universal modular ACTOR formalism for artificial intelligence, 1973, Morgan Kaufmann Publishers Inc.
- [Jaf16] A. Jafari, Performance Evaluation and Model Checking of Probabilistic Real-time Actors, Reykjavíi University, 2016.
- [LZG+87] E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik. Quantitative system performance: computer system analysis using queueing network models, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1987
- [MA03] P. D. Manuel, J. AlGhamdi. A Data-centric Design for N-tier Architecture, Inf. Sci. Inf. Comput. Sci., 150(3-4), 2013, pp. 195-206.
- [Mar04] M. Marzolla. Simulation-Based Performance Modeling of UML Software Architectures, 2004.
- [Rob00] T. G. Robertazzi. Computer Networks and Systems. Queueing Theory and Performance Evaluation, Springer-Verlag New York, 2000, 409 pages.
- [PMT01] K. Goseva-Popstojanova, A. P. Mathur, K. S. Trivedi. Comparison of architecture-based software reliability models, Software Reliability Engineering, 2001, pp. 22-31.
- [SBC17] G. Serazzi, M. Bertoli, G. Casale. Java Modelling Tools. User Manual. 2017. [Žiūrėta 2017-05-07]. Prieiga per internetą:
<http://jmt.sourceforge.net/Papers/JMT_users_Manual.pdf>.
- [TRM14] The reactive manifesto, 2014. [Žiūrėta 2017-05-05]. Prieiga per internetą:
<<http://www.reactivemanifesto.org>>.

- [TKL+11] A. B. Todd, A. K. Keller, M. C. Lewis, M. G. Kelly, Multi-agent System Simulation in Scala: An Evaluation of Actors for Parallel Simulation, PDPTA 2011: Proceedings of the 2011 International Conference on Parallel and Distributed Processing Techniques and Applications, 1, 2011, pp. 162-168.
- [Ver15] V. Vernon. Reactive Messaging Patterns with the Actor Model: Applications and Integration in Scala and Akka, 2015.