VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS

Formalių specifikacijų taikymas projektuojant paskirstytas sistemas

Applying Formal Specifications to Design Distributed Systems

Magistro darbo planas

Atliko:	Matas Savickis	(parašas)
Darbo vadovas:	Karolis Petrauskas, Doc., Dr.	(parašas)
Recenzentas:	Viačeslav Pozdniakov, Partn. Doc.	(parašas)

Vilnius – 2020

TURINYS

1. Įvadas

Šiais laikais daugumas didelių programinių sistemų yra kuriamos paskirstytų sistemų principu[**mcr**]. Šioms sistemoms plečiantis ir didėjant atsiranda problema, kaip efektyviai ir patikimai paskirstyti gaunamus duomenis sistemoje. Vienas iš būdų tai padaryti yra naudoti paskirstytų srautinių duomenų platformomis. Viena iš populiariausių šiuo metu yra Apache Kafka platforma.

Apache Kafka buvo pradėta kurti kompanijos LinkedIn siekiant sukurti centralizuotą įvykių valdymo platformą skirtą interneto duomenų integravimo užduotims atlikti. 2012 metai Apache Kafka sistema buvo perduota Apache Software Foundation tolesniam vystymui. Šiuo metu Apache Kafka platforma yra publikuoti-prenumeruoti architektūros žinučių siuntimo sistema, kurios dizainas pasižymi lengvu plečiamumu, patvarumu, patikimumu ir greičiu. Duomenys Apache Kafka platformoje yra išsaugomi saugiu, trukdžiams atspariu būdu. Apache Kafka kūrėjų teigimu, šiuo metu platforma naudoja daugiau negu 80 procentų didžiausių Jungtinių Valstijų įmonių. Kafka platforma yra plačiai naudojama įvairiose srityse tokiuose kaip žurnalistika, debesijos paslaugos, muzikos srauto paslaugos, telekomunikacijos, bankinės paslaugos ir daugelis kitų.

Norint užtikrinti, kad Apache Kafka vartotojai būtų patenkinti platformos kokybe kūrėjais yra implementavę skirtingų testų siekdami užtikrinti platformos kokybę. Testai padeda atskleisti programines klaidas arba pasakyti ar naujai parašytas kodas nepaveikė seniau parašyto funkcionalumo. Tačiau net ir laikantis gerųjų testavimo praktikų nepavyksta išvengti programinių klaidų. Net ir paskyrus daugiau resursų testavimui, netrivialiuose sistemose, tokiose kaip Apache Kafka, pilnas sistemos testavimas yra neįmanomas. Todėl norint atrasti subtilesnius sisteminius sutrikimus tenka naudoti kitus metodus. Vienas iš tokių metodų yra formalios specifikacijos.

Formalios specifikacijos yra matematikės technikos skirtos apibūdinti kuriamų sistemų elgseną ir padėti kuriant jos dizainą naudojant griežtas ir veiksmingas priemones. Turint formalią specifikaciją galima pasinaudoti ka parodant, kad sistemos dizainas yra korektiškas pagal sukurtą specifikaciją. Tai leidžia koreguoti sistemos dizainą dar prieš pradedant jį implementuoti, taip išvengiant didelių piniginių investicijų, kaip blogi sistemos dizaino sprendimai paaiškėtų vėlesniuose sistemos kūrimo stadijose. Formalios specifikacijos sudaromos pasinaudojant tam tikra kalba arba įrankiu. Viena iš tokių formalaus specifikavimo kalbų yra TLA$^+$.

TLA$^+$ yra formalios specifikacijos kalba sukurta Leslie Lamport. Leslie Lamport 1980 metais sukūrė laiko veiksmų logiką(angl. Temporal Logic of Actions) pasinaudodamas Amir Pnueli 1997 metais sukurta laiko logika(angl. Temporal Logic). 1999 metais Leslie Lamport, pasinaudodamas savo sukurta laiko veiksmų logika, sukūrė formalaus specifikavimo kalbą TLA$^+$. TLA$^+$ kalba yra skirta kurti konkurencinių ir paskirstytų sistemų formalios specifikacijoms ir šias specifikacijas verifikuoti. Kadangi TLA$^+$ specifikacija yra parašyta formalia kalba tai leidžia sukurti modelius, kurių elgesį galima patikrinti pažingsniui ir stebint ar modelis nepažeidžia apibrėžto nekintamumo. TLA$^+$ specifikacijos buvo sėkmingai taikytos ir industrijoje.

Viena iš šiuo metu žinomiausių formalių specifikacijų ir TLA$^+$ panaudojimo industrijoje sėkmės istorijų yra Amazon Web Service(AWS) komandos 2014 metais išleistas straipsnis. Straipsnyje rašoma, kad AWS komanda TLA$^+$ formalias specifikacijas panaudojo dešimtyje didelių projektų ir AWS tuo metu turėjo 7 komandas, kurios naudojo TLA$^+$ kurdamos naujus programinius sprendi-

mus. AWS sistemos specifikavimo metu buvo surasti 10 iki šiol neatrastų sisteminių klaidų, kurių atradimas ir pasiūlyti ištaisymai atskleidė tolimesnes sistemines klaidas. Straipsnyje įvardinta ir kita, netiesioginė nauda gauti formaliai specifikuojant sistemą: pagerėjęs bendros sistemos suvokimas, padidėjęs produktyvumas ir inovacijos.

Dar viena sėkmės istorija yra 2018 metais Kafka Summit konferencijoje pristatyta Apache Kafka TLA$^+$ formali specifikacija sukurta Jason Gustafson. Pristatyme buvo parodyta, kad pritaikius TLA$^+$ specifikuojant Apache Kafka duomenų replikavimo algoritmą buvo surastos ir pataisytos 3 retais atsitinkančios programinės klaidos. Keletas programinių klaidų buvo surastos ir pataisytos taisant jau minėtas tris pradines klaidas.

2. Temos aktualumas bei naujumas

Iki šiol, kiek mums žinoma, Apache Kafka platforma buvo specifikuota tik vieną kartą neakademiniame kontekste ir sukurta specifikacija atnešė naudos padėdama surasti sistemines klaidas. Panašią mokslininkų sėkmę matome ir Amazon Web Service formalios specifikacijos sudarymo tyrimuose. Dėl papildomų Apache Kafka formalių specifikacijų stokos ir praeityje pasisekusio formalaus specifikavimo paskirstytose sistemose manome, kad papildomi tyrimai Apache Kafka platformoje atneštu naudos surandant sisteminių klaidų arba užtikrinant, kad specifikuotoje sistemos dalyje jų nėra. Šiuo metu Apache Kafka sisteminių klaidų registre yra išspręstų ir neišspręstų klaidų kurių verifikavimas padėtų atskleisti naujas klaidas arba įrodyti kad klaidos ištaisytos korektiškai. Apache Kafka platforma turi daug naudotojų, todėl tolimesnis kokybės užtikrinimas Apache Kafka platformoje atneštų naudą. Sėkmingai identifikavus paskirstytų sistemų architektūrinius šablonus būtų galima įrodyti korektiškumą daug didesnei paskirstytų sistemų aibei ir šio darbo rezultatais būtų galima vadovautis kuriant atitinkamas paskirstytas sistemas.

3. Darbo tikslas

Pagerinti Apache Kafka platformos kokybę surandant sisteminių klaidų arba įrodant, kad sukurtoje specifikacijoje klaidų nėra. Įrodyti didesnės paskirstytų sistemų aibės algoritmų korektiškumą specifikuojant architektūrinius šablonus.

4. Uždaviniai

1. Išnagrinėti literatūrą susijusią su formaliais metodais, TLA$^+$ specifikavimo kalba bei Apache Kafka patlforma.
2. Išskirti Apache Kafka platformos modulius, kurie bus formaliai specifikuoti.
3. Specifikuoti išskirtas Apache Kafka platformos dalis naudojant TLA$^+$ specifikavimo kalbą.
4. Įvertinti sukurtos formalios specifikacijos korektiškumą.
5. Įvertinti, ar pagal sukurtą specifikaciją Apache Kafka platforma veikia korektiškai.
6. Esant poreikiui įrodyti specifikacijos teoremas.
7. Patikrinti ar Apache Kafka implementaciją atitinka specifikaciją.
8. Apache Kafka platformoje surasti architektūrinius šablonus, kurie yra taikoki ir kituose paskirstytuose sistemose ir juos specifikuoti.

5. Laukiami rezultatai

1. Pasirinktų Apache Kafka modulių specifikacija.
2. Įrodymas apie specifikacijos korektiškumą.
3. Apache Kafka parašytos specifikacijos ir implementacijos sutapimo įvertinimas.
4. Išskirti ir specifikuoti paskirstytų sistemų šablonai taikomi kitose platformose.