TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kristjan Pint IABB178985

Modulaarsete andmemudelite simuleerimine SAP HANA andmebaasi haldussüsteemis

Bakalaureusetöö

Juhendajad: Toomas Klementi
Magistrikraad

Jakob Jõgi
Magistrikraad

Tallinn 2021

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kristjan Pint

18.05.2021

Annotatsioon

Antud töö eesmärk oli välja selgitada autori töökohas arendatavale monoliitsele süsteemile võimalikud alternatiivid ja analüüsida nende jõudlust suurte andmehulkade juures.

Autor valis võimalikeks alternatiivideks kolm arhitektuurilist lahendust ja lõi nendel lahendustel põhinevad rakendused. Võimekuse hindamiseks loodi nendele rakendustele simulatsioonid, mis jäljendavad päriselu.
Töö tulemusena saadi andmed loodud süsteemide võimekuse kohta ja leiti nendest parim.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 43 leheküljel, 6 peatükki, 20 joonist, 45 tabelit.

Abstract

Modular Data Model Simulations in SAP HANA Database Management System

The aim of this work was to find out possible alternatives to the monolithic system developed in the author's workplace and to analyze their performance during large data volumes.

The author chose three architectural solutions as possible alternatives and created applications based on each one of them.To assess the performance of these applications, the author created simulations that mimic real life. As a result of the work, data on the capabilities of the created systems was obtained and the best of them was found.

The thesis is in Estonian and contains 43 pages of text, 6 chapters, 20 figures, 45 tables.

Lühendite ja mõistete sõnastik

API	(Application Program Interface):Rakendusliides-reeglid ja vahendid rakendusprogrammi suhtluseks.
AWS	Amazon Web Services
CRUD	(C-create, R- read, U-update, D-delete) - Andmebaasidega seotud põhioperatsioonid
Deploy	Metoodiline protseduur protsessi, meetodi, toote, teenuse kasutuselevõtuks organisatsiooni kõigis asjakohastes kohtades.[4]
Gateway	Funktsionaalüksus, mis rajab tee läbi ühe või mitme arvutivõrgu.[4]
HDI	Hana Deployment Infrastructure
MTA	Multi Target Application
SQL	Deklaratiivne interaktsiooni- ja programmikeel relatsioonandmebaaside halduseks ja kasutamiseks[4]

Sisukord

1 Sissejuhatus 11
2 Kirjanduse ülevaade 13
2.1 Andmebaas **13**
2.2 Relatsioonandmebaas **13**
2.3 SAP HANA **13**
2.4 Cloud Foundry ja HANA XSA **13**
2.5 Konteinerid ja SAP HDI konteinerid **14**
2.6 Andmebaasi sünonüüm **15**
2.7 Tarkvara arhitektuur **15**
2.8 Monoliitne tarkvara arhitektuur **15**
2.9 Mikroteenustel põhinev jagatud tarkvara arhitektuur **16**
3 Varasemad uuringud 17
4 Metoodika 18
4.1 Kasutatud tehnoloogiad: **18**
4.2 Arhitektuuri tüüpide valik **18**
4.2.1 Monoliitne arhitektuur 18
4.2.2 Jagatud arhitektuur 18
4.3 Andmemudel **19**
4.4 Andmebaaside tehnilised lahendused **22**
4.5 Serveripoolsete rakenduste tehnilised lahendused **23**
4.6 Simulatsioon **24**
4.6.1 Tehniline kirjeldus 24
4.6.2 Simulatsiooni tsüklite töö kirjeldus 24
4.6.3 Simulatsioonide läbiviimine 25
5 Tulemused ja analüüs 26
5.1 Tulemused **26**
5.1.1 Sünonüümidel põhineva arhitektuuri simulatsioonide tulemused 26
5.1.2 Mikroteenustel põhineva arhitektuuri simulatsioonide tulemused 31
5.1.3 Monoliidil põhineva arhitektuuri simulatsioonide tulemused 35
5.2 Analüüs **39**
5.3 Edasiarendamise võimalused **41**

6 Kokkuvõte 42
7 Viited 43
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks 44
Lisa 2: TABELID 45

Jooniste loetelu

Joonis 1. Cloud foundry organisatsiooni struktuur 14
Joonis 2. Andmemudel 20
Joonis 3. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(sünonüümid 50k) 27
Joonis 4. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(sünonüümid 50k) 27
Joonis 5. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(sünonüümid 100k-1) 28
Joonis 6. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(sünonüümid 100k-1) 29
Joonis 7. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(sünonüümid 100k-2) 30
Joonis 8. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(sünonüümid 100k-2) 30
Joonis 9. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(mikroteenused 50k) 31
Joonis 10. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(mikroteenused 50k) 32
Joonis 11. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa(mikroteenused 100k-1) 33
Joonis 12. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(mikroteenused 100k-1) 33
Joonis 13. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(mikroteenused 100k-2) 34
Joonis 14. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(mikroteenused 100k-2) 35
Joonis 15. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(modulaarne monoliit 50k-1) 36
Joonis 16. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(modulaarne monoliit 50k) 36
Joonis 17. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa(modulaarne monoliit 100k-1) 37
Joonis 18. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(modulaarne monoliit 100k-1) 37
Joonis 19. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(modulaarne monoliit 100k-2) 38
Joonis 20. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(modulaarne monoliit 100k-2) 39

Tabelite loetelu

Tabel 1. Simulatsioonis tekkinud kirjete arvud (sünonüümid 50k) 26
Tabel 2. Simulatsiooni kokkuvõtte (sünonüümid 50k) 26
Tabel 3. Simulatsioonis tekkinud kirjete arvud(sünonüümid 100k-1) 28
Tabel 4. Simulatsiooni kokkuvõte (sünonüümid 100k-1) 28
Tabel 5. Simulatsioonis tekkinud kirjete arvud(sünonüümid 100k-2) 29
Tabel 6. Simulatsiooni kokkuvõte (sünonüümid 100k-2) 29
Tabel 7. Simulatsioonis tekkinud kirjete arvud(mikroteenused 50k) 31
Tabel 8. Simulatsiooni kokkuvõte (mikroteenused 50k) 31
Tabel 9. Simulatsioonis tekkinud kirjete arvud(mikroteenused 100k-1) 32
Tabel 10. Simulatsiooni kokkuvõte (mikroteenused 100k-1) 32
Tabel 11. Simulatsioonis tekkinud kirjete arvud(mikroteenused 100k-2) 34
Tabel 12. Simulatsiooni kokkuvõte (mikroteenused 100k-2) 34
Tabel 13. Simulatsioonis tekkinud kirjete arvud(modulaarne monoliit 50k) 35
Tabel 14. Simulatsiooni kokkuvõte (modulaarne monoliit 50k) 35
Tabel 15. Simulatsioonis tekkinud kirjete arvud(modulaarne monoliit 100k-1) 37
Tabel 16. Simulatsiooni kokkuvõte (modulaarne monoliit 100k-1) 37
Tabel 17. Simulatsioonis tekkinud kirjete arvud(modulaarne monoliit 100k-2) 38
Tabel 18. Simulatsiooni kokkuvõte (modulaarne monoliit 100k-2) 38
Tabel 19. Olemite loetelu 45
Tabel 20. Olem element 45
Tabel 21. Olem parameter 45
Tabel 22. Olem elementparameter 46
Tabel 23. Olem activity 46
Tabel 24. Olem product 46
Tabel 25. Olem workunit 47
Tabel 26. Olem producttransaction 47
Tabel 27. Olem acitvitytransaction 47
Tabel 28 simulatsiooni kokkuvõte olemite kaupa(sünonüümid 50k) 48
Tabel 29. Simulatsiooni kokkuvõte päringu tüübi kaupa(sünonüümid 50k) 48
Tabel 30. Simulatsiooni kokkuvõte olemite kaupa(sünonüümid 100k-1) 49
Tabel 31. Simulatsiooni kokkuvõte päringu tüübi kaupa(sünonüümid 100k-1) 49
Tabel 32. Simulatsiooni kokkuvõte olemite kaupa(sünonüümid 100k-2) 50
Tabel 33. Simulatsiooni kokkuvõte päringu tüübi kaupa(sünonüümid 100k-2) 50
Tabel 34. Simulatsiooni kokkuvõte olemite kaupa(mikroteenused 50k) 51
Tabel 35. Simulatsiooni kokkuvõte päringu tüübi kaupa(mikroteenused 50k) 51
Tabel 36. Simulatsiooni kokkuvõte olemite kaupa(mikroteenused 100k-1) 52
Tabel 37. Simulatsiooni kokkuvõte päringu tüübi kaupa(mikroteenused 100k-1) 52
Tabel 38. Simulatsiooni kokkuvõte olemite kaupa(mikroteenused 100k-2) 53
Tabel 39. Simulatsiooni kokkuvõte päringu tüübi kaupa(mikroteenused 100k-2) 53
Tabel 40. Simulatsiooni kokkuvõte olemite kaupa(modulaarne monoliit 50k) 54
Tabel 41. Simulatsiooni kokkuvõte päringu tüübi kaupa(modulaarne monoliit 50k) 54

Tabel 42. Simulatsiooni kokkuvõte olemite kaupa(modulaarne monoliit 100k-1) 55
Tabel 43. Simulatsiooni kokkuvõte päringu tüübi kaupa(modulaarne monoliit 100k-1).....55
Tabel 44. Simulatsiooni kokkuvõte olemite kaupa(modulaarne monoliit 100k-2) 56
Tabel 45. Simulatsiooni kokkuvõte päringu tüübi kaupa(modulaarne monoliit 100k-2).....56

1 Sissejuhatus

Tarkvaraarenduses on pikka aega olnud peamiseks viisiks rakenduste loomisel arendada need ühise tükina. Selline lähenemine tagab arendajatele kindla kontrolli süsteemi töö üle.
Maailm on liikumas aina enam andmestunud ühiskondade juurde. See tähendab, et kasutatakse ja talletatakse aina rohkem informatsiooni meie ümber toimuvast. Ühise tükina arendatud rakendustel on sellises maailmas tekkinud kitsaskohad. Kuigi need rakendused on oma iseloomult lihtsa ülesehitusega, võivad need aina suurema informatsiooni hulga käitlemisel paisuda hallatamatuteks. Selle lahendusena on kanda kinnitamas lahendus, kus rakendusi luuakse jaotatud tükkidena, kus igal tükil on kanda oma roll. Neid tükke on süsteemides võimalik välja vahetada. Seega, isegi kui oma olemuselt on tükkidest rakenduste ehitamine keerulisem, võimaldavad need arendajatel andmete käitlemiseks luua kõikvõimalikke lahendusi.

Ka autori töökohas on tekkinud probleem, kus ühe tükina arendatavas rakenduses saab aina enam ilmsiks, et andmete haldamine on muutunud keeruliseks. Seetõttu on jõutud punkti, kus tuleks otsustada, kuidas edasi liikuda. Kas peaks leidma kiiresti arenevas andmemudelis võimaluse suure andmehulga jätkusuutlikuks käitlemiseks või tuleks süsteem jagada eraldi tükkidesse.

Käesoleva töö eesmärgiks on välja selgitada võimalikud alternatiivid ja analüüsida nende jõudlust suurte andmehulkade juures.
Autor arvas, et parimaks lahenduseks oleks süsteemi jagamine erinevatesse tükkidesse.

Tulemuseni jõudmiseks on autoril plaanis valida kolm võimalikku lahendust, realiseerida nende põhjal ettevõttes kasutatava andmemudeli vähendatud versioonid ja analüüsida nende võimekust. Võimekuse hindamiseks loob autor igale rakendusele simulatsioonid, mis jäljendavad päriselu kasutust.

Loodetava tulemusena saab autor loodud simulatsioonide põhjal vastu võtta otsuse, millist lahendust tuleks kasutada.

Töö algab sissejuhatusega. Seejärel viiakse läbi ülevaade olemasolevast kirjandusest. Praktilise poole pealt antakse kõigepealt ülevaade kasutatud tehnoloogiatest, peale mida tutvustatakse realiseeritavate lahenduste arhitektuure. Tehnilise lahenduste presenteerimiseks kirjeldatakse kasutatav andmemudel, realiseeritud andmebaaside lahendused, serveripoolsete rakenduste lahendused ning loodud simulatsioonid. Töö tulemused koosnevad simulatsioonidest kogutud andmetest ja nende analüüsist.

2 Kirjanduse ülevaade

2.1 Andmebaas

Andmebaasid on oma olemuselt väga triviaalsed. Andmebaasidesse salvestatakse mingil kindlal viisil informatsiooni, et seda hiljem kasutada. Kui kuskile on mingil organiseeritud kujul andmed talletatud, siis on tegemist andmebaasiga. Andmebaas ei pea ilmtingimata olema arvutiprogramm. Tarkvarateaduses mõeldakse andmebaasi all tavaliselt mingit programmi, milles on võimalik andmeid talletada, neid pärida ja vastavalt vajadusele muuta. [8]

2.2 Relatsioonandmebaas

Relatsioonandmebaas ehk seoste andmete kogu töötab kasutajate jaoks kindlat sorti andemete sidumisega, mida tarkvarateaduses võetakse kui andmetabelit. Relatsioonandmebaaside puhul ei küsita andmeid nende kindla asukoha järgi, vaid kindlaks määratud seoste põhjal. Seoste hoiustamise tõttu ei ole relatsioonandmebaaside puhul vajalik teada salvestatud andmete reaalset asukohta. See lubab hoida andmeid ka üksteisest eraldatuna. [8]

2.3 SAP HANA

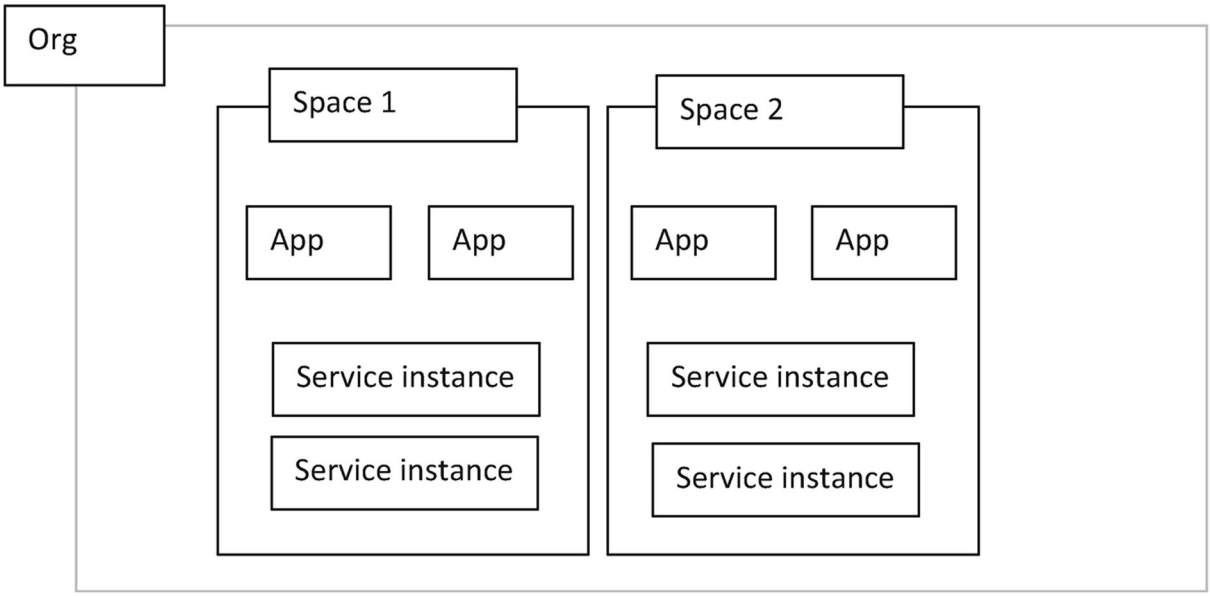
SAP HANA on platvorm mis sisaldab mälusisest relatsioonilist andmebaasi, veebiserverit ja muid mootoreid, mis moodustavad kõrge saadavuse ja jõudlusega analüütilise platvormi.
SAP HANA puhul on laiemas üldsuses levimas kaks arvamust. SAP HANA peetakse kas veerupõhist andmete hoiustamist kasutavaks andmebaasiks või analüütiliseks mootoriks, mis aitab ärilisi otsuseid vastu võtta. Tegelikult on SAP HANA nendest funktsionaalsustest kokku pandud platvorm, mida saab ettevõtetes kasutada nii andmebaasina kui ka analüütilise mootorina.[6, 7] Antud töös kasutatakse SAP HANA andmebaasina.

2.4 Cloud Foundry ja HANA XSA

Cloud Foundry on platvorm, mille peal on võimalik arendada mitmes erinevas programmeerimiskeeles rakendusi. Lisaks tagab Cloud Foundry ka arendatavate lahenduste pilves töötamise.[5] Kui rakendus on arendatud Cloud Foundry platvormile, siis on ta sisuliselt eraldatud muu maailmaga seotud infrastruktuurist. Cloud Foundry valmidusega rakendusi on võimalik väga lihtsalt liigutada näiteks arvutist ettevõtte serverisse või siis mõne pilveteenuse pakkuja serverisse. Cloud Foundry's on defineeritud organisatsioon(organization) – ruum(space) – rakendus(application) hierarhiline struktuur. Organisatsioon on defineeritud

arenduskontona, kus ühel või mitmel arendajal on määratud rollid, mis võimaldavad neil rakendusi arendada või teistele ligipääsu lubada. Ühes organisatsioonis võib olla üks või mitu ruumi, mida kasutatakse rakenduste loomiseks, ülalhoiuks või jooksutamiseks. Rakendus (ja teenused) võimaldavad kasutajatel mingeid andmeid näha ja kasutada [6]

SAP HANA XSA(Extended Application Services) on Cloud Foundry'l põhinev platvorm, millele on tehtud SAP'i infrastruktuuriga liitmiseks muudatused ja lisad. Lisadeks on näiteks integratsioon SAP HANA andmebaasiga. [12]



Joonis 1. Cloud Foundry organisatsiooni struktuur [6, lk 11]

2.5 Konteinerid ja SAP HDI konteinerid

Konteineri all mõeldakse tarkvarateaduses lahendust, mis lubab arendajatel oma rakendust pakkida üheks, teistest eraldi töötavaks, üksususeks. Konteinerid töötavad tavaliselt mõnel serveril ja jagavad serverile paigaldatud riistvara. Kuna kõik konteinerid on serveril eraldiseisvad, võimaldab see rakenduste kiiret ja tõhusat välja laskmist.[3]

SAP HDI on teenus, mis võimaldab arendajatel andmebaase pakkida konteineritesse. HDI teenus sisaldab kõiki peamisi SAP HANA andmebaasi omadusi.[11]

Igale HDI konteinerile luuakse tehnilised kasutajad. Läbi tehniliste kasutajate toimub konteinerite haldus, andmete pärimine ja manipuleerimine.

2.6 Andmebaasi sünonüüm

Andmebaasi sünonüüm on andmeobjekt, mis võimaldab viidata näiteks tabelile, mis võib eksisteerida lokaalsel või välisel serveril.

SAP HANA andmebaasi haldussüsteemis on sünonüüm ainus lahendus andmeobjektide ligipääsu tagamisel üle andmebaasi skeemi. SAP HANA kontekstis on andmebaasi skeem ja konteiner üsna tihedalt seotud.

2.7 Tarkvara arhitektuur

Süsteemi tarkvara arhitektuur on kogus struktuure, mis tagavad ühe süsteemi töö.

Nende struktuuride peale on ehitatud ülesse kogu rakenduse töö. Igal rakendusel on arhitektuur, sõltumata sellest, kas see on planeeritud või mitte. Tarkvara struktuurid koosnevad programmidest, funktsionaalsusest ja nende vahelisest suhtlusest ning mõlema tunnustest.

Struktuurid saab jaotada kolme gruppi:

- Komponent ja ühendaja struktuurid, mis on fokuseeritud funktsionaalsete tükkide suhtlusele süsteemi töös.
- Moodul struktuurid, mis jagavad süsteemid implementeeritavatesse üksustesse.
- Määravad struktuurid, mis kaardistavad tarkvara struktuuride allutamise mittetarkvaralistele üksustele ehk riistvarale.

Tarkvara arhitektuuri kohta käib printsiip, et iga tarkvara süsteem luuakse mingi ettevõtte äriliste vajaduste saavutamiseks ja süsteemi arhitektuur ühendab ärivajadused loodava tarkvaraga. Tarkvara arhitektuuri disainimine ja analüüsimine aitab kaasa ärivajaduste saavutamiseks.[2]

2.8 Monoliitne tarkvara arhitektuur

Monoliitsel tarkvara arhitektuuril ehitatud rakendused on ehitatud ühe tervikuna ja koosnevad tavaliselt kolmest osast: kliendipoolne kasutajaliides, andmebaas ja serveri poolne rakendus. Serveri poolses rakenduses tehakse kõik vajalikud toimingud: tehakse kasutajale valmis veebilehed, vajalikud loogilised arvutused ja andmebaasi päringud.

Monoliitne lahendus on kõige naturaalsem lähenemine süsteemi ehitamisel ja ilmselt alustavad kõik arendajad oma karjääri just monoliitsete rakenduste ehitamisega.

Monoliitsete rakenduste üheks plussiks ja samal ajal ka miinuseks on süsteemi terviklikkus. Kogu rakendus tuleb luua ja käivitada ühe üksusena. Muudatuste korral tuleb välja lasta rakendusest uus versioon ja välja vahetada kogu töötav rakendus. [10]

2.9 Mikroteenustel põhinev jagatud tarkvara arhitektuur

Mikroteenused on tarkvara maailmas kiiresti populaarsust koguv võimalik lähenemine, kus rakendused luuakse mitmest väiksest eraldiseisvast funktsionaalsest osast. Nende üksuste vahel on suhtlus lahendatud tihti kasutades veebipäringuid. See lähenemine võimaldab rakendustele suurema paindlikkuse. Kõik rakenduse osad luuakse eraldi, seega saab neid ka vajadusel jooksvalt vähendada ja juurde lisada.[10]

3 Varasemad uuringud

Mikroteenused on viimaste aastate jooksul muutunud väga populaarseks, tänu sellele on nende kohta väga palju informatsiooni. Välja on toodud erinevad lahenduste viisid, nende plussid ja miinused.
Monoliitsed rakendused on olnud kasutusel juba väga pikka aega, seetõttu leidub piisavalt informatsiooni ka monoliitsel arhitektuuril põhinevate rakenduste kohta.
Samuti on palju informatsiooni selle kohta, kuidas liikuda üle monoliitselt arhitektuurilt mikroteenustel põhinevale arhitektuurile.

Monoliitse ja mikroteenustel põhineva arhitektuuri jõudlusmomenti käsitlevatest töödest parimaks näiteks peab autor tööd ”Microservices: Granularity vs. Performance”.
Antud tööst tuleb välja, et kuigi mikroteenused pakuvad oma olemuselt paindlikke süsteeme, siis saadud paindlikkuse hinnaks on langenud jõudlus. Langenud jõudlus tekib, kuna teenuste vahelisel suhtlusel kasutatakse veebipäringuid. Mikroteenustel ehitatud süsteemides aeglustub käitluskiirus sõltuvalt veebipäringute ülekandekiirustest. [9]
Kuna autor kasutab andmebaaside haldamiseks eraldi konteinereid, siis teemaga lähemalt tutvumiseks kasutas autor tööd “Performance Evaluation of Microservices Architectures Using Containers”. Antud töös on välja toodud jõudlusmoment erinevatel serverilahendustel põhineval rakenduse skaleerimisel. [1]
SAP HANA-ga seotud küsimuste tekkimisel kasutati peamiselt SAP HANA XSA ametlikku dokumentatsiooni.[12]
Antud töö uudsus ja vajadus on tingitud olukorrast, kus erinevatel arhitektuuri tüüpidel lahendatud rakendused luuakse SAP HANA platvormile, sellelaadset tööd autoril leida ei õnnestunud. Lisaks on töös kasutatud ühe lahendusena erinevate teenuste ühendamiseks sünonüüme. Sünonüüme kasutatakse tavaliselt ühes andmebaasis asuvate tabelite ühendamisel. SAP HANA platvormis on võimalik sünonüüme kasutada erinevate konteinerite vahel, kuid selliselt lahendatud süsteemi võimekuse analüüsi autoril leida ei õnnestunud.

4 Metoodika

Antud töö tulemuste saamiseks lõi autor kolm simulatsiooni, millega taheti välja selgitada erinevatel arhitektuuri tüüpidel põhinevate rakenduste jõudlus suurte andmehulkade ja pideva andmevoo juures. Loodavatele süsteemidele olid seatud järgmised piirangud:

- Andmebaas peab olema modulaarne
- Süsteemis peab olema kajastatud moodulite vaheline andmevahetus
- Süsteemis peab toimuma töö suure hulga andmetega
- Süsteemis peab toimuma pidev andmevoog

4.1 Kasutatud tehnoloogiad:

Töös kasutati SAP HANA andmebaasi haldussüsteemi. Arenduskeskkond oli üles seatud HANA XSA organisatsioonis. Kuna autoril ei olnud võimalik kasutada mitmeid SAP HANA instantse, kasutati ühes ruumis loodud HDI konteinereid, mida kasutati erinevatel aegadel. Erinevate rakenduste käitamiseks kasutati JavaScript'il põhinevat Node.js käitlussüsteemi. Rakendusliideste ja simulatsioonide programmeerimisel kasutati JavaScript programmeerimiskeelt. Andmete töötlemisel kasutati struktuuripäringukeelt SQL. HTTP päringute tegemisel kasutati Node.js paketti node-fetch. Veebipäringute käitlemiseks kasutati raamistikku Express.js. Päringutes ja vastustes kasutatav andmeformaat on JSON. Andmemudeli disainimisel kasutati vabavaralist tarkvara Draw.io.

4.2 Arhitektuuri tüüpide valik

4.2.1 Monoliitne arhitektuur

Kuna ühe piiranguna oli ette antud, et süsteemi andmebaas peab olema modulaarne, siis lõi autor ühe süsteemina, parema termini puudumisel, Modulaarse Monooliitrakenduse. Rakenduses on küll kolm andmebaasi konteinerit, kuid süsteem on kasutatav vaid ühe tervikuna

4.2.2 Jagatud arhitektuur

Autor lõi kaks jagatud arhitektuuriga süsteemi. Süsteemides on jagatus lahendatud erinevalt. Ühes süsteemis on suhtlus teenuste vahel lahendatud rakendusliideses kasutatavate HTTP päringutega ja teises kasutades sünonüüme.

HTTP päringute kasutamine on saanud standardseks mikroteenuseid kasutavate rakenduste teenuste vahelise suhtlemise tagamisel, seetõttu valis autor selle üheks lahenduseks.
Kuna SAP HANA andmebaasi haldussüsteemis on võimalik HDI konteinerite ja ka erinevate SAP HANA instantsi HDI konteinerite vaheline andmete ühendamine sünonüümidega, siis valis autor jagatuse tagamise üheks lahenduseks andmebaasi sünonüümid.

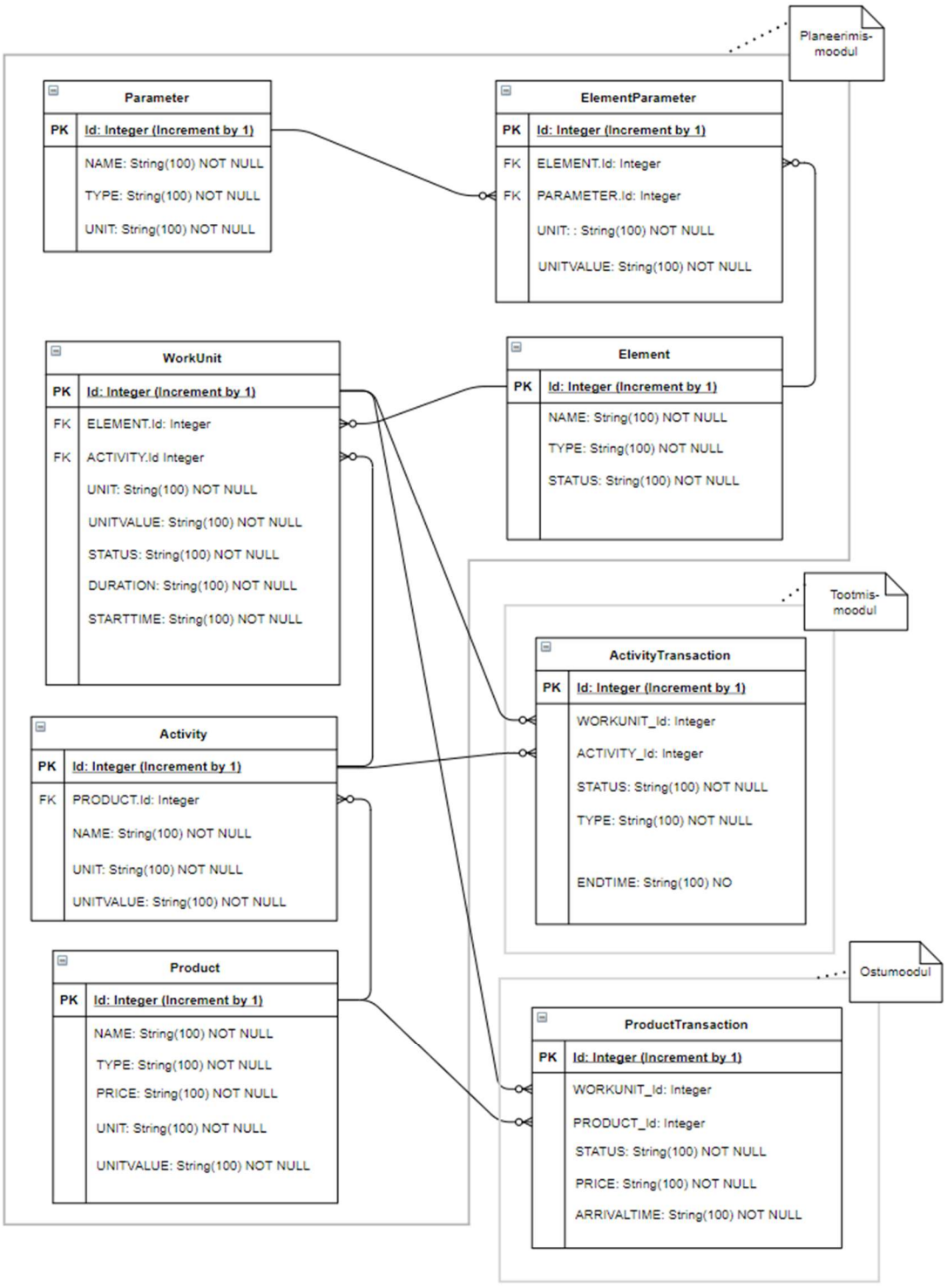
4.3 Andmemudel

Kuna SAP HANA üheks osaks on mälusisene relatsiooniline andmebaas, siis loodi andmebaasi kavandamiseks relatsiooniline andmemudel.

Rakenduste kavandamisel võeti aluseks autori töökohas sätestatud vajadused:

- Andmemudel peab olema modulaarne.
- Andmemudelis peab tekkima suures mahus andmeid.
- Andmemudelis peab olema tagatud andmete moodulite vaheline edasi-tagasi suhtlus

Andmemudeli aluseks võeti autori töökohas kasutusel oleva äriprotsessi miniatuurne ja lihtsustatud versioon. Mudeliga jäljendatakse ehitiste 3D mudeli realiseerimist ehitusprojektides. Kogu rakenduse andmevoogu kirjeldavasse andmemudelisse luuakse kolm moodulit, millest planeerimismoodul on peamoodul ja ostumoodul ning tootmismoodul alammoodulid. Kuna simulatsiooni jaoks kavandatav andmemudel ei olnud keeruline, lõi autor ainult andmemudeli füüsilise disaini.
Andmemudeli graafilise esitluse loomisel kasutati “varese jala”(ing. *crow-foot*) notatsiooni ja tehti olem-seose diagramm.



Joonis 2. Andmemudel

Süsteemis on kasutusel kolm moodulit: planeerimise moodul, ostumoodul ja tootmismoodul ning import skriptid. Planeerimise moodulis kasutatakse importitud ja moodulis genereeritavaid ning tekitatavaid andmeid. Olemeid WorkUnit ja Activity kasutatakse kõigis moodulites. Olemit Product kasutatakse planeerimise ja ostumoodulis.

Olemite kirjeldused:

Element kirjeldab ehitise 3D mudelis planeeritud objekti. Kuna antud töös Elemente 3D mudelist ei loeta, siis genereeritakse elemendid süsteemis. Olemil Element on seos olemiga ElementParameter.
Parameter kirjeldab ühe elemendi võimalikke parameetreid ehk omadusi. Süsteemi luuakse 10 parameetrit import skriptiga. Parameetrite kirjeid süsteemi töö käigus ei muudeta. Olemil Parameter on seos olemiga ElementParameter.
ElementParameter on olemite Element ja Parameter vaheline seos. ElementParameter'ite väärtustega on võimalik arvutada elemendi realiseerimiseks vajaminevate tööde ja toodete kulu. Kui päriselt tulevad ElementParameter kirjete väärtused 3D mudelist, siis antud töös omastatakse neile parameetrite väärtused.
Activity ehk tegevus kirjeldab ühe elemendi realiseerimiseks vajalikku tegevust. Süsteemi luuakse 10 tegevust import skriptiga. Reaalsuses võib ühe elemendi realiseerimiseks vaja minna rohkem kui ühte tegevust, kuid antud töös luuakse igale elemendile üks tegevus. Activity'l on seos olemitega WorkUnit ja ActivityTransaction.
Product kirjeldab ühe tegevusega kaasnevat toodet. Olemil Product on seos olemitega Activty ja ProductTransaction.
WorkUnit ehk tööühik on olemite Element ja Activity seos, mis kirjeldab ühe elemendi realiseerimiseks planeeritavat tööd.
ProductTransaction on ühe WorkUnit'i realiseerimiseks tarvis vajamineva toote ostu kujutav kirje. ProductTransaction'id luuakse ostumoodulis.
ActivityTransaction kirjeldab ühe elemendiga kaasnevate tööde tegemist. ActivityTransaction kirjed luuakse tootmismoodulis.

4.4 Andmebaaside tehnilised lahendused

Antud töös luuakse kolm süsteemi, mis on arhitektuuriliselt lahendatud erinevatel viisidel. Sellega kaasnevalt on ka süsteemide andmebaaside ühenduste loomised mõnevõrra erinevad. Andmebaaside konteinerid luuakse SAP HANA's rakenduse andmebaaside moodulite ehitamisel(ing. *build*). Selle töö käigus luuakse konteiner, andmebaasi skeem, tabelid, tehnilised kasutajad jms. Kuna selle automaatse tekkimise kirjeldus ei ole antud töö skoobis, siis seda rohkem ei analüüsitud. Konteineritega ühendused defineeritakse rakenduse development descriptor(arenduse kirjeldus) failis mta.yaml. Failis mta.yaml defineeritakse rakenduses kasutusel olevad moodulid, milleks antud töös on rakendusliides ja andmebaasi konteiner või konteinerid ning defineeritakse vajalikud ressursid. Failis mta.yaml defineeritud konteineritega luuakse ühendus rakenduse algfailis(asukoht defineeritud mta.yaml'is, antud töös server.js). Sünonüümide kasutamisel algfailis sünonüümidele eraldi ühendust looma ei pea.

Sünonüüme kasutava süsteemi andmebaasi lahenduse juures oluline fakt, et Igal SAP HDI konteineril on vähemalt 2 tehnilist rolli. Kaheks peamiseks rolliks on andmeobjekti(konteineri) omanik ja andmeobjekti kasutaja(mitte segamini ajada tavakasutajaga).
Sünonüümide kasutamiseks tuleb SAP HANA's luua vajalike õigustega tehniliste kasutajate rollid. Rollidega on võimalik määratleda, kas ja kellel on võimalik konteinerit kasutada ja mis päringuid teha lubatakse. Antud töös rollidega tehniliste kasutajate õigusi ei piiratud ning tagati moodulite vahel täies mahus kasutatavus. Rollid defineeriti planeerimise moodulis ja neid kasutati ostu- ja tootmismoodulis sünonüümidele õiguste andmisel.
Konteineri omanikule loodi lubamise õigusega roll tagamaks ligipääs teistele objektidele ja õigus jagada ligipääsud teistele kasutajatele.
Andmeobjekti kasutajatele loodi roll kasutusõiguste loomiseks. Andmeobjekti kasutajale peab andma loa võõraste objektide kasutamiseks andmeobjekti omanik.
Sünonüümide kasutamiseks tuleb alammoodulis sünonüüm defineerida, teha vajalikud konfiguratsioonid ja anda talle ligipääsuks vajalikud õigused. Ostumoodulis kasutati läbi sünonüümide tabeleid "Activity","Product" ja "WorkUnit". Tootmismoodulis kasutati läbi sünonüümide tabeleid "Activity" ja "WorkUnit".

Päringuid kasutavas süsteemis loodi kolm eraldi rakendust, kus konteinerid ja andmebaasi ühendused tehti peatüki alguses välja toodud standardlahendusega. Erinevate moodulite vaheline andmete suhtlus tagati rakenduste vaheliste päringutega.

Modulaarse monoliitrakenduse lahenduses loodi kolm andmebaasi konteinerit ja üks rakendus. Rakendusele anti ressurssidena kõikide andmebaaside ühendused.

Kolm olemit: Parameter, Product ja Activity on püsivate väärtustega. Erandiks on Activity, mille kirjetele lisatakse simulatsiooni töö alguses võõrvõtmena ühe toote identifikaator. Sellest hoolimata otsustas autor nende kolme olemi andmed süsteemidesse luua kasutades import skripti. Import skriptid käivitakse planeerimise mooduli käivitamisel või modulaarse monoliitrakenduse käivitamisel.

4.5 Serveripoolsete rakenduste tehnilised lahendused

Päringute käitlemiseks kasutati serveripoolsetes rakendustes standardseid CRUD meetodeid. Parema struktuuri hoidmiseks loodi igale olemile teenus(mitte segi ajada mikroteenusega), milles defineeriti õige tabel, millega see konkreetne teenus töötas.
Andmete õigsuse kontrollid(ing. *integrity check*) loodi teenustele, millega seotud andmetabelite kirjetel on seosed teiste tabelite kirjetega. Need loodi, sest üle konteinerite ei ole võimalik andmeid siduda võõrvõtmetega. Õigsuse kontrollides veenduti, kas viite taga eksisteerib ka reaalne kirje.

Kuna modulaarses monoliitrakenduses oli kasutusel mitu andmebaasi konteinerit, siis tuli teenustele, millel oli vaja kasutada andmeid erinevatest konteineritest, anda ligipääs mitmele konteinerile. Autor lahendas selle marsruutijas, kus anti teenuse meetodite sisendiks ühe andmebaasi konteineri liidestuse asemel kahe konteineri liidestused.

Mikroteenustel põhinevatel süsteemidel kasutati mikroteenustena süsteemi erinevaid mooduleid.
Kui HTTP päringutel põhineval mikroteenuste süsteemide juures kasutatakse standardina päringu õige mikroteenuse juurde suunamiseks tavaliselt gateway’d, siis autor ei pidanud süsteemi lihtsuse tõttu seda vajalikuks. Autor defineeris mikroteenuses väliste olemite

teenused, samamoodi CRUD meetoditega, mille sisuks oli õige mikroteenuse pihta õige päringu tegemine. Sisuliselt töötab see lahendus staatilise gateway'na.
Sünonüüme kasutava süsteemi teenustes kasutati vajadusel tabeli definitsiooni asemel sünonüümi.

4.6 Simulatsioon

4.6.1 Tehniline kirjeldus

Antud töös loodud simulatsioon töötab neljas tsüklis: genereerimise tsükkel, planeerimise tsükkel, ostutsükkel ja tootmistsükkel. Simulatsiooni parameetriteks on simulatsiooni kestvus, ühe täisringi aeg ja genereeritavate elementide vahemik. Ühe täisringi jooksul läbitakse genereerimise tsüklit neli korda, planeerimise tsüklit viis korda, ostu tsüklit kolm korda ja tootmise tsüklit kaks korda. Tsüklite läbimisega tahtis autor tekitada olukorda, kus ühe ehitise 3D mudeli reaalsuseks saamiseks tuleb kõige rohkem muuta plaane, seejärel tuleb antud plaanide alusel teha ostuordereid vajalike toodete soetamiseks ning protsessis kõige vähem kordi tehakse reaalselt ehitustöid.

Simulatsiooni läbimine toimub asünkroonselt, simulatsioon ei oota tsüklite sisu lõppemist järgmisesse tsüklisse liikumiseks. Tsüklite sisu on sünkroonne, kuna tsüklite sees tehtavad toimingud võivad sõltuda tsüklis tehtud päringutest. Simulatsioonis kogutakse andmeid päringute kiiruse kohta. Kogutavateks andmeteks on päringu URL, päringu timestamp, päringu tulemuse ridade arv, päringu suurus baitides, päringu tegemiseks kulunud aeg millisekundites, kuupäev ja päringu meetod.

4.6.2 Simulatsiooni tsüklite töö kirjeldus

Genereerimise tsüklis luuakse süsteemi parameetrina etteantud vahemikus juhuslik arv juhuslike väärtustega elemente ja salvestatakse need planeerimise mooduli andmebaasi.
Planeerimise tsüklis lisatakse mittesalvestatud staatusega elementidele juhuslik arv parameetreid ja salvestatakse need seosed andmebaasi ElementParameter'itena. Mittesalvestatud staatusega elementidest, salvestatud elementparameetritest ja tegevustest luuakse WorkUnit'id ehk tööühikud. Tööühikud luuakse vaid elementidest, millel on seos parameetritega, mille ühikuks on 'mm' või 'PC'. Nende parameetrite seast valitakse üks

juhuslik parameeter. Valitud juhusliku parameetri ühiku väärtusest sõltuvalt valitakse selle ühikuga sobivate tegevuste seast juhuslik tegevus.

Loodavatele tööühikutele määratakse ühikuks ja ühikuväärtuseks valitud tegevuse ühiku ja ühikuväärtuse väärtused. Töö kestvuse arvutamiseks jagatakse parameetri väärtus(õiges astmes) tegevuse ühiku väärtusega.

Tööühikutele lisatakse andmekirje mahu suurendamiseks juhuslik kuupäev. Tööühiku staatuseks määratakse 'CONFIRMED' ehk kinnitatud. Kui elemendile luuakse tööühik, siis määratakse elemendi staatuseks 'SAVED' ehk salvestatud. Kui elemendile tööühikut ei looda, siis määratakse elemendi staatuseks 'REMOVED' ehk eemaldatud. Tööühikute valmimisel salvestatakse nende väärtused andmebaasi. Peale tööühikute salvestamist uuendatakse andmebaasis elementide väärtused. Peale vajalike päringute tegemist arvutatakse fiktiivne graafik ja eelarve, kus on kajastatud kogumaksumus, kogu aeg, maksumus ja kogus toote kohta ning aeg ja kogus tegevuse kohta.

Ostu tsüklis luuakse kinnitatud tööühikutele ProductTransaction'id ehk ostuorderid nende tööde tegemiseks vajalike toodete kohta. Peale ostu ostuorderite loomist ja salvestamist uuendatakse kasutatud tööühikute staatused väärtuseks 'READY' ehk tööühik on tootmisvalmiduses.

Tootmise tsüklis luuakse tootmisvalmiduses tööühikutele tootmist või paigaldust tähistavad ActivityTransaction'id ehk tegevus transaktsioonid. Tegevus transaktsioonide staatuseks määratakse 'STARTED' ehk tegevus on alustatud. Tegevus transaktsioonide tüübiks määratakse 'ASSEMBLY' ehk kokkupanek.

4.6.3 Simulatsioonide läbiviimine

Tulemuste saamiseks viis autor läbi iga süsteemi kohta kolm simulatsiooni. Simulatsiooni kestus oli kolm tundi ning ühe täisringi pikkus kümme minutit. Esimeses simulatsioonis valiti genereerimise tsüklis loodavate elementide hulgaks 49990 kuni 50000. Teises ja kolmandas simulatsioonis valiti loodavate elementide hulgaks 99990 kuni 100000. Simulatsioonid viidi läbi öösiti või nädalavahetuseti, vältimaks teiste XSA organisatsioonis olevate kasutajate mõju süsteemile. Simulatsioonis kasutatav SAP HANA instants asus AWS Iirimaal asuvas virtuaalserveris. Peale igat simulatsiooni koguti andmebaasi konteineritest vajalikud andmed ning seejärel kustutati nende sisu, vältimaks olukorda, kus simulatsiooni tulemusi mõjutavad varem loodud andmed. Simulatsioonide käigus koguti andmed eraldi csv failidesse.

5 Tulemused ja analüüs

5.1 Tulemused

5.1.1 Sünonüümidel põhineva arhitektuuri simulatsioonide tulemused

SÜNONÜÜMID 50k

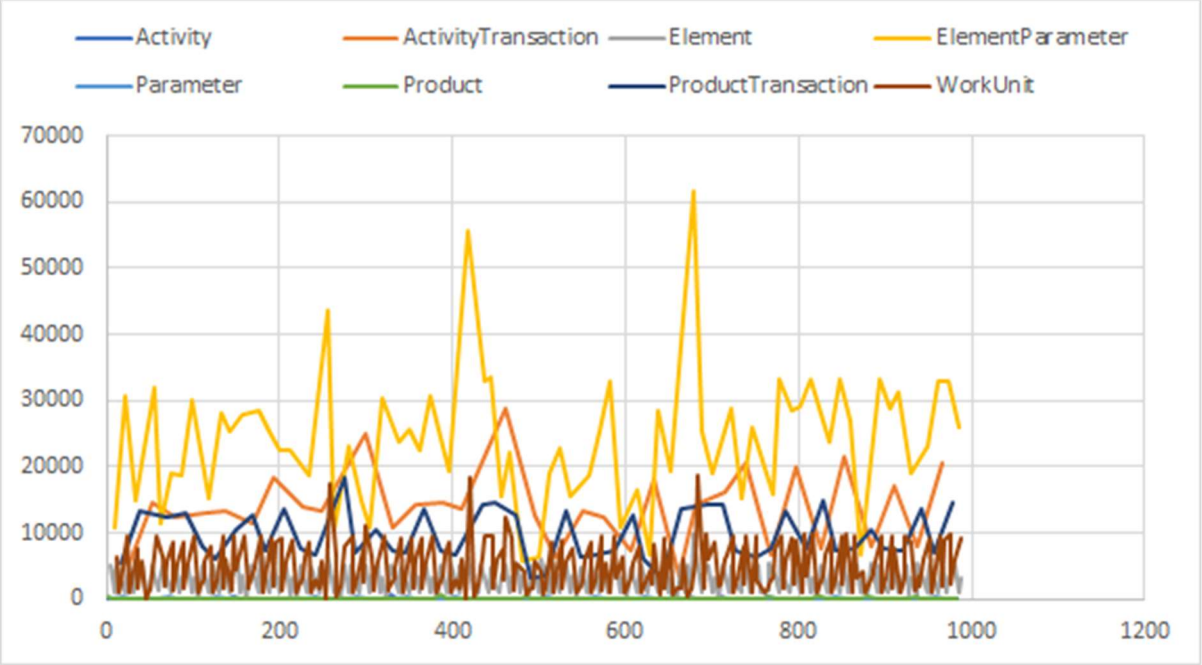
Genereeritud elementide hulk ühes tsüklis: 49990-50000. Kokku tehtud päringuid 988. Täielik andmete ülevaate Lisas 2 tabelites 28 ja 29. Simulatsioonis märgata kolm ElementParameter'i salvestamisel ilmunud ekstreemumit.

Olem	tekkinud kirjeid
Element	3599645
ElementParameter	15073309
WorkUnit	3157414
ProductTransaction	3109767
ActivityTransaction	3010746

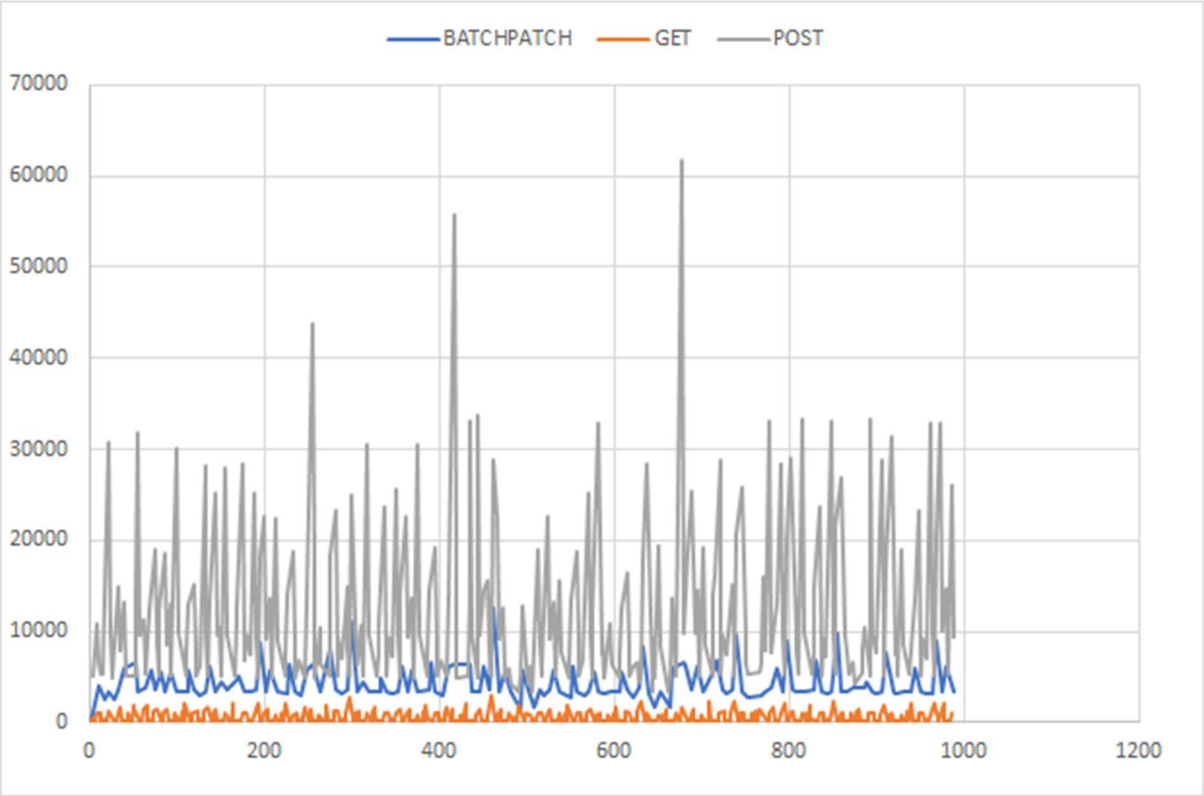
Tabel 1. Simulatsioonis tekkinud kirjete arvud (Sünonüümid 50k)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	min	max	avg	min	max	avg	min	max	avg	min	max
48274	0	569420	5044470	11	49777741	4586	197	61774	1302	0.02	9587

Tabel 2. Simulatsiooni kokkuvõtte (Sünonüümid 50k)



Joonis 3. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(Sünonüümid 50k)



Joonis 4. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(Sünonüümid 50k)

SÜNONÜÜMID 100k-1

Genereeritud elementide hulk ühes tsüklis: 99990-100000. Kokku tehtud päringuid 789. Simulatsiooni töö lõpetati enneaegselt. Täielik andmete ülevaate Lisas 2 tabelites 30 ja 31.

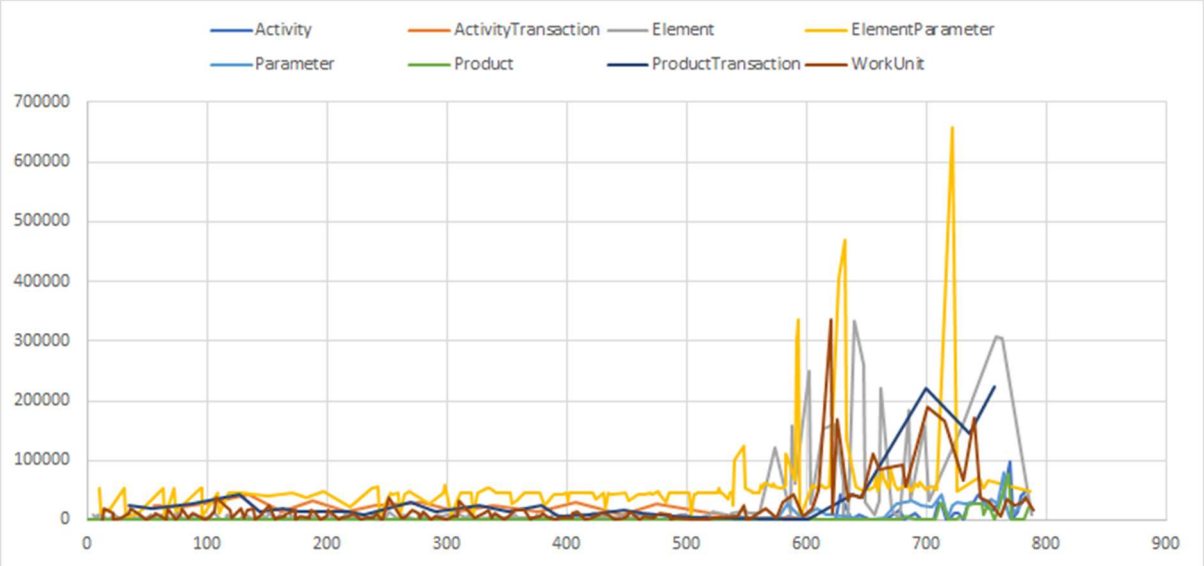
Selgelt tuleb välja punkt, kus simulatsioon satub raskustesse. ElementParameter'ite salvestamisele kuluv aeg hakkas eksponentsiaalselt kasvama, sest simulatsiooni planeerimise tsüklid jõudsid üksteisele järgi ja veel muutmata staatusega elementidele hakati looma iga kord ElementParameter'eid.

Olem	tekkinud kirjeid
Element	6399685
ElementParameter	19774454
WorkUnit	2906621
ActivityTransaction	2534843
ProductTransaction	2975738

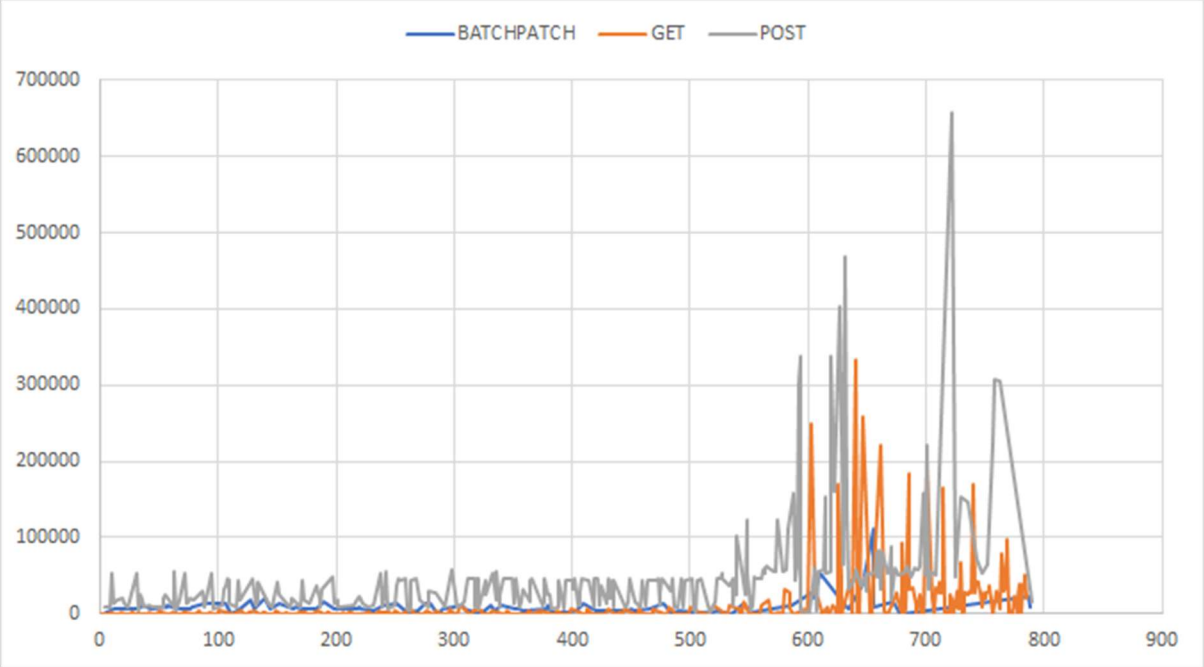
Tabel 3. Simulatsioonis tekkinud kirjete arvud(Sünonüümid 100k-1)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	min	max	avg	min	max	avg	min	max	avg	min	max
78401	0	1365780	7363269	11	94694330	24786	196	658580	908	0.0003	10287

Tabel 4. Simulatsiooni kokkuvõte (Sünonüümid 100k-1)



Joonis 5. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(Sünonüümid 100k-1)



Joonis 6. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(Sünonüümid 100k-1)

SÜNONÜÜMID 100k-2

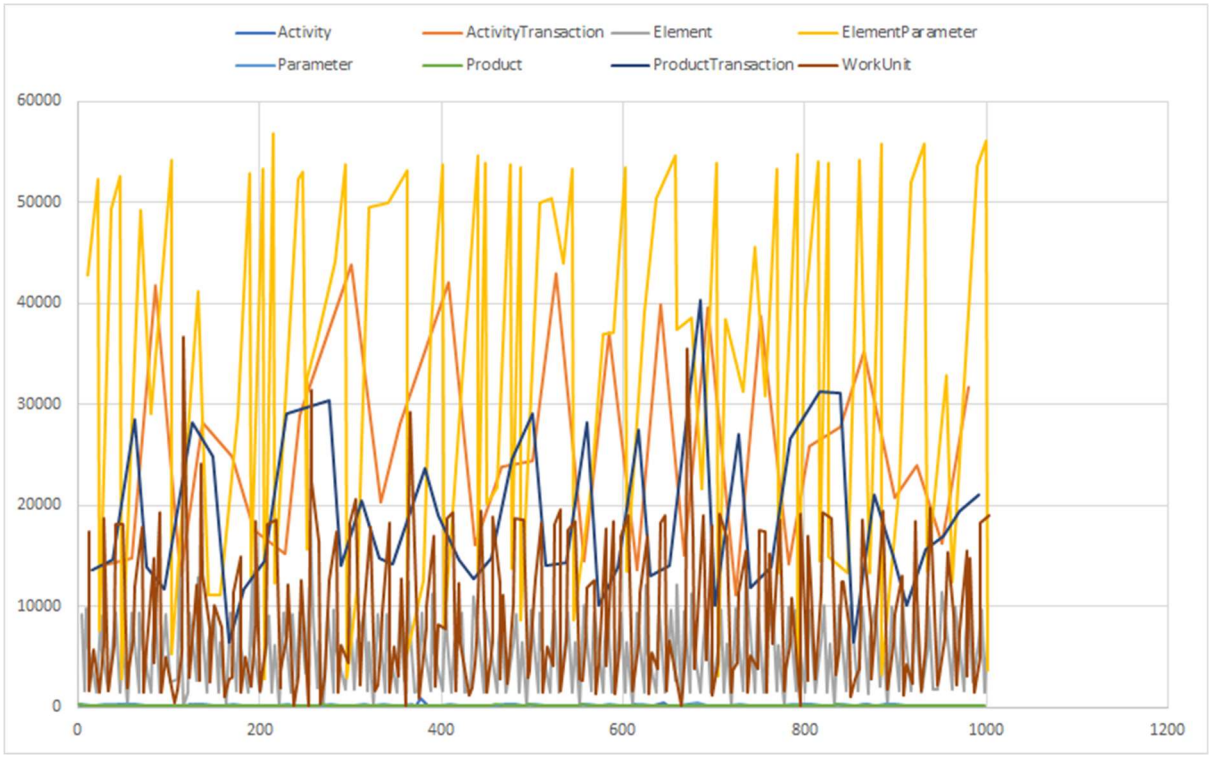
Genereeritud elementide hulk ühes tsüklis: 99990-100000. Kokku tehtud päringute arv 1003. Täielik ülevaade Lisas 2 tabelites 32 ja 33. Simulatsioon läbiti stabiilselt, selgeid erijuhte ei tekkinud. Selgelt näha erinevate olemite käitlemiseks kuluv ressurss.

Element	7199661
ElementParameter	28626891
WorkUnit	5936066
ActivityTransaction	5665389
ProductTransaction	5743635

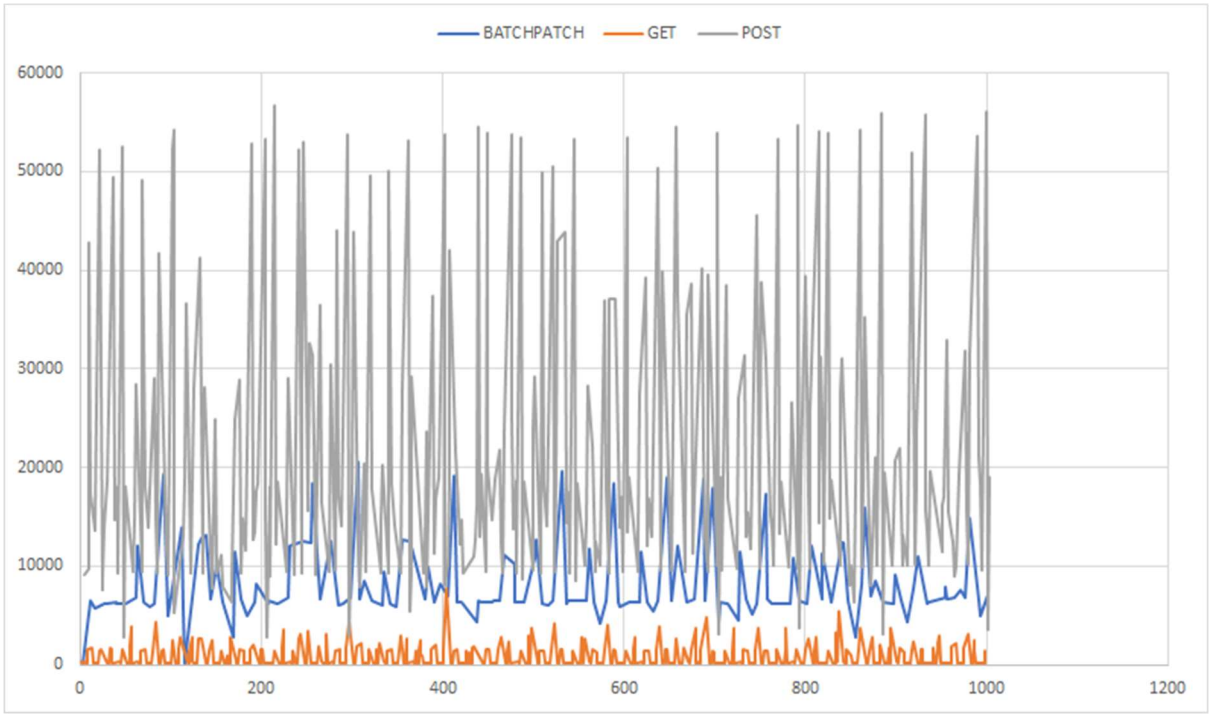
Tabel 5. Simulatsioonis tekkinud kirjete arvud(Sünonüümid 100k-2)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	min	max	avg	min	max	avg	min	max	avg	min	max
91518	0	500000	9542339	11	44208274	8394	196	56754	1494	0.001	10481

Tabel 6. Simulatsiooni kokkuvõte (Sünonüümid 100k-2)



Joonis 7. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(Sünonüümid 100k-2)



Joonis 8. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(Sünonüümid 100k-2)

5.1.2 Mikroteenustel põhineva arhitektuuri simulatsioonide tulemused

MIKROTEENUSED 50k

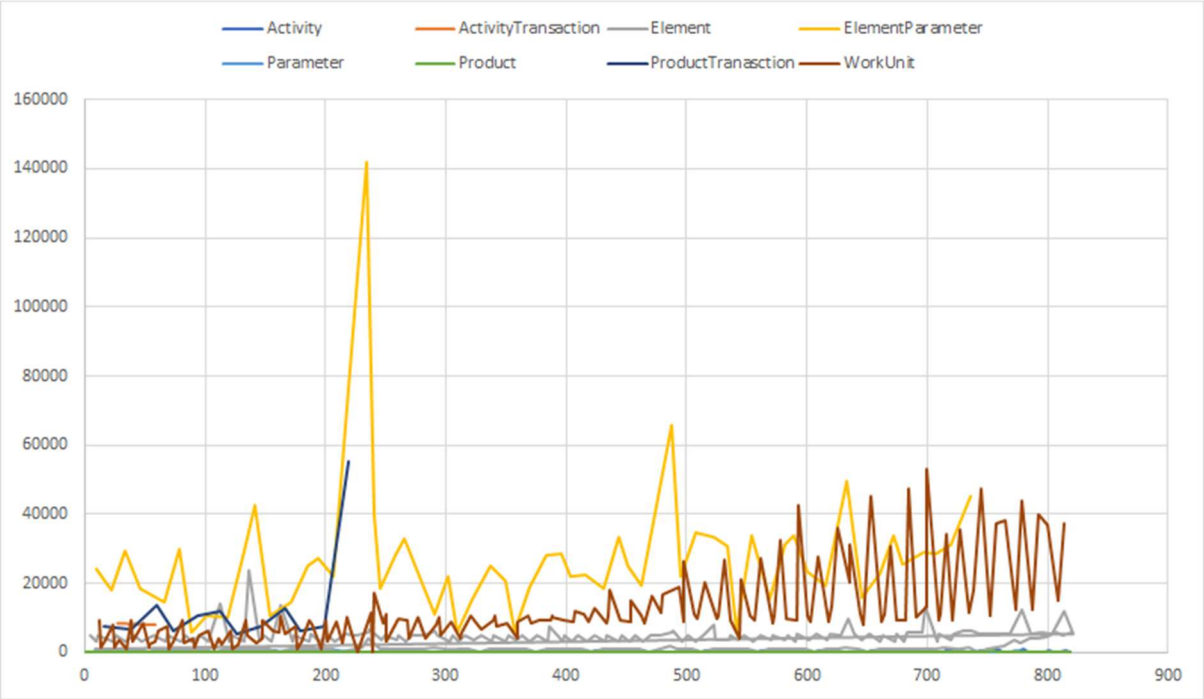
Genereeritud elementide hulk ühes tsüklis: 49990-50000. Kokku tehtud päringute arv 1003. Täielik ülevaade Lisas 2 tabelites 33 ja 34. Selgelt nähtav erijuht simulatsiooni algusfaasis, mil üks ElementParameter'i salvestamise päring toimus normaalsest üle kolme korra aeglasemalt. Erijuhu põhjus teadmata. Simulatsiooni lõppfaasis märgata päringutele kuluva aja pikenemist.

Element	3599633
ElementParameter	12632381
WorkUnit	2666015
ProductTransaction	666539
ActivityTransaction	91062

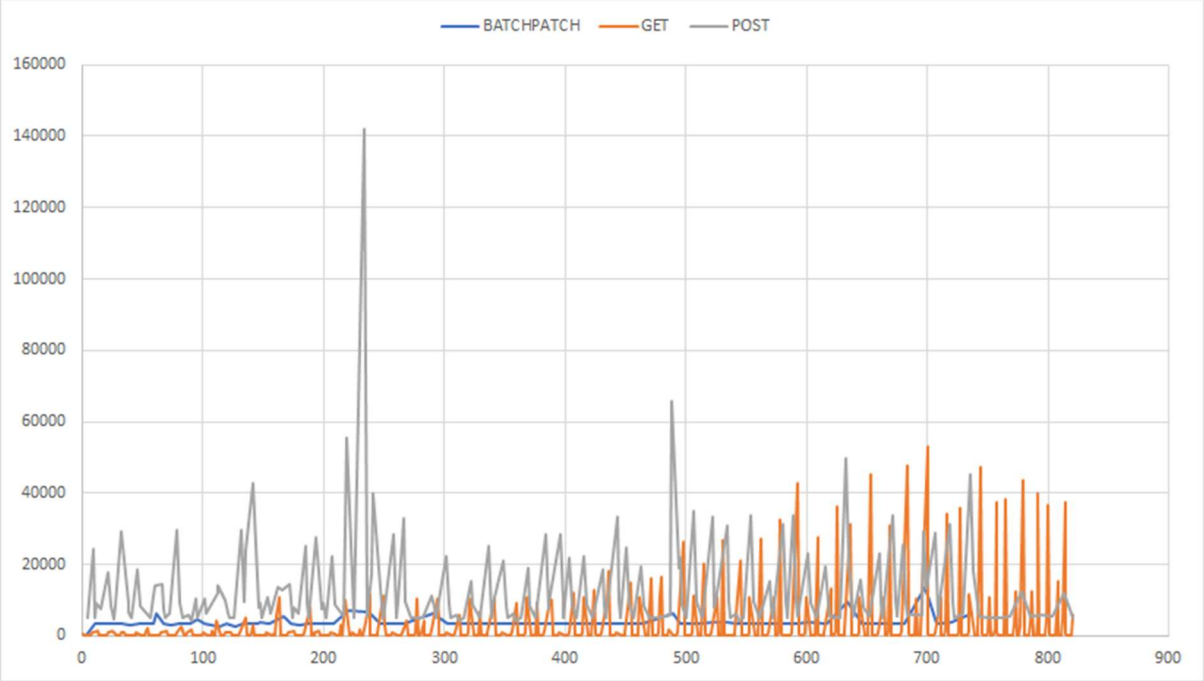
Tabel 7. Simulatsioonis tekkinud kirjete arvud(Mikroteenused 50k)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	min	max	avg	min	max	avg	min	max	avg	min	max
114118	0	1999476	14610002	1092	286736668	5477	196	141890	1434	0.001	8184

Tabel 8. Simulatsiooni kokkuvõte (Mikroteenused 50k)



Joonis 9. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(Mikroteenused 50k)



Joonis 10. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(Mikroteenused 50k)

MIKROTEENUSED 100k-1

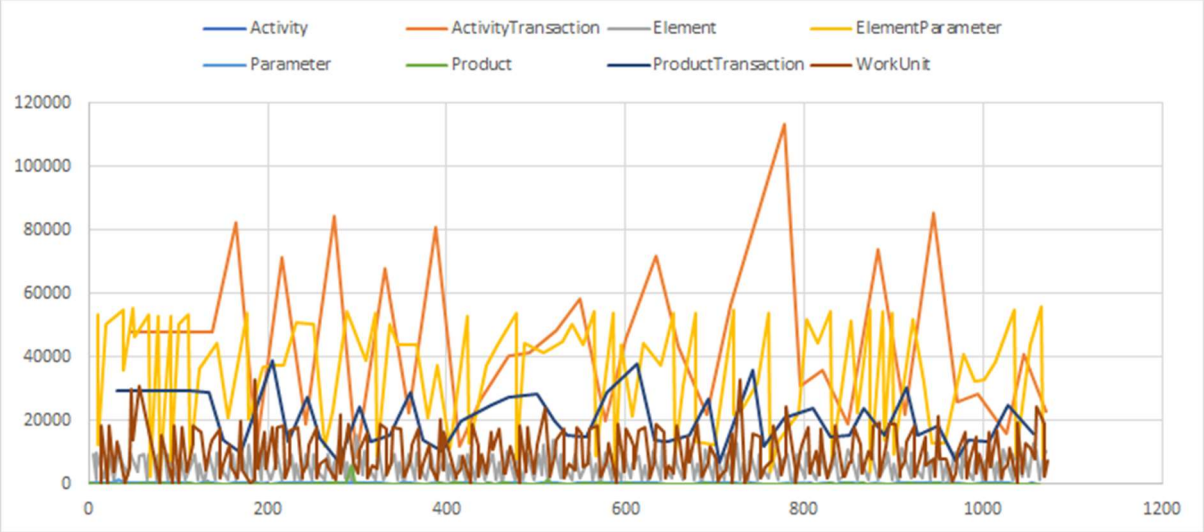
Genereeritud elementide hulk ühes tsüklis: 99990-100000. Kokku tehtud päringute arv 1072. Täielik ülevaade Lisas 2 tabelites 36 ja 37. Simulatsiooni töö sujus stabiilselt. Märgata ActivityTransaction'ite käitlemisel kulunud suuremat ajakulu.

Element	7299669
ElementParameter	26461277
WorkUnit	5968640
ProductTransaction	5742105
ActivityTransaction	5742105

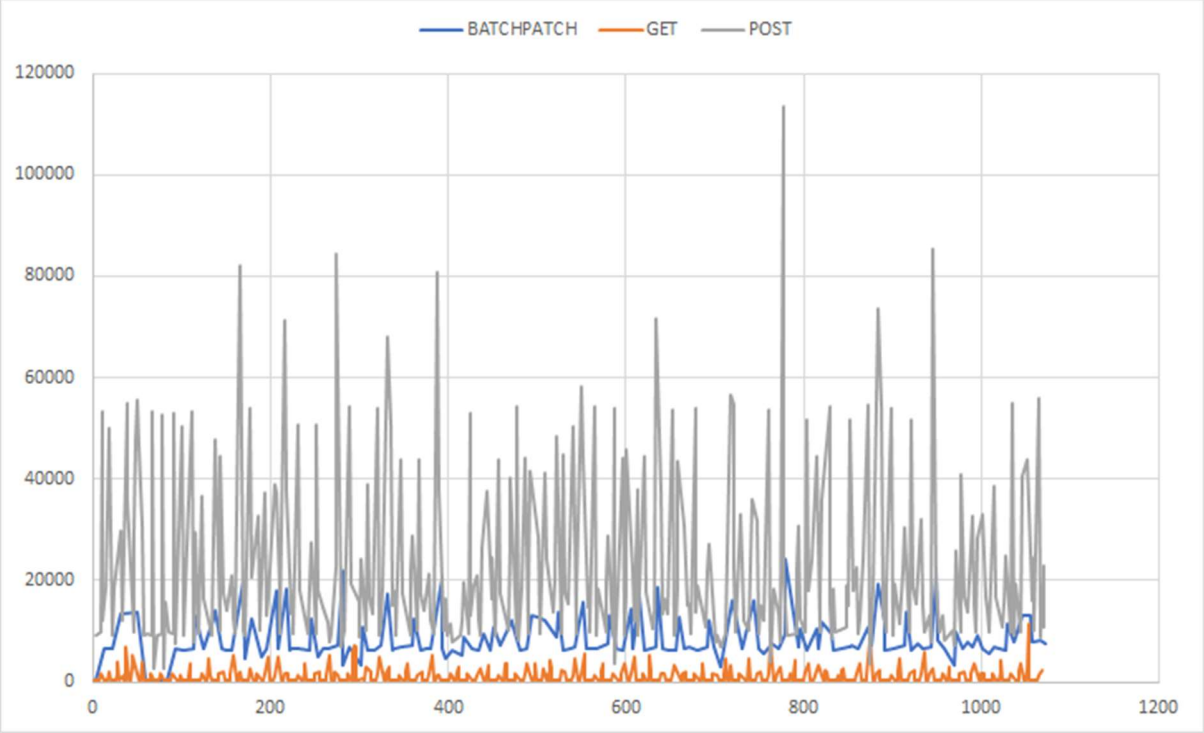
Tabel 9. Simulatsioonis tekkinud kirjete arvud(Mikroteenused 100k-1)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	min	max	avg	min	max	avg	min	max	avg	min	max
90322	0	500000	10205915	11	52211147	9126	196	113483	1365	0.001	7918

Tabel 10. Simulatsiooni kokkuvõte (Mikroteenused 100k-1)



Joonis 11. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa(Mikroteenused 100k-1)



Joonis 12. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(Mikroteenused 100k-1)

MIKROTEENUSED 100k-2

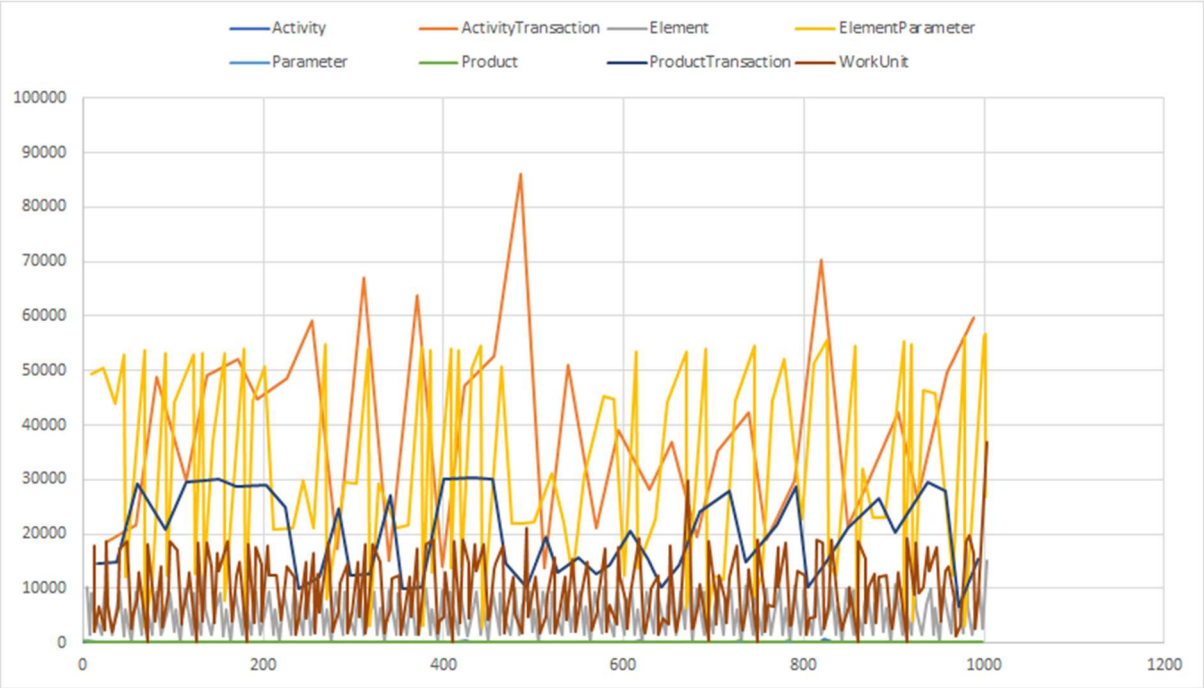
Genereeritud elementide hulk ühes tsüklis: 99990-100000. Kokku tehtud päringute arv 1004. Täielik ülevaade Lisas 2 tabelites 38 ja 39. Simulatsiooni töö sujus stabiilselt.

Element	7199660
ElementParameter	27090920
WorkUnit	5889011
ActivityTransaction	5593699
ProductTransaction	5690953

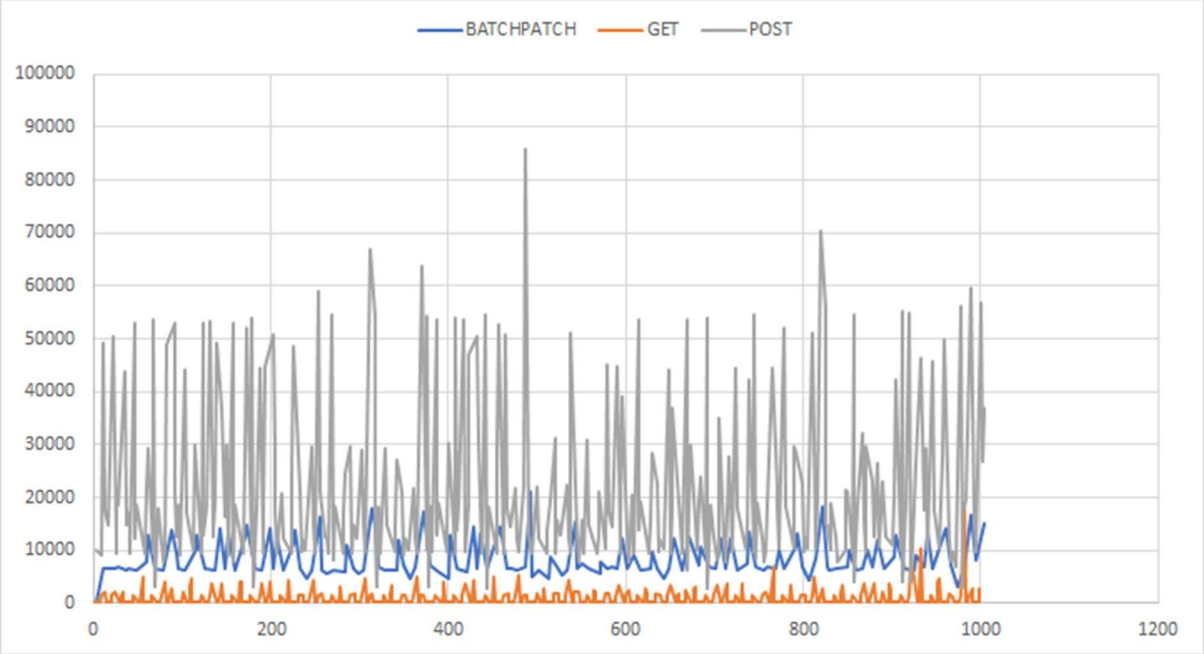
Tabel 11. Simulatsioonis tekkinud kirjete arvud(Mikroteenused 100k-2)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	max	min	avg	max	min	avg	max	min	avg	max	min
88283	500000	0	10079131	52210873	11	8837	86001	196	1397	7871	0.03

Tabel 12. Simulatsiooni kokkuvõte (Mikroteenused 100k-2)



Joonis 13. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(Mikroteenused 100k-2)



Joonis 14. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(Mikroteenused 100k-2)

5.1.3 Monoliidil põhineva arhitektuuri simulatsioonide tulemused

MODULAARNE MONOLIIT 50k

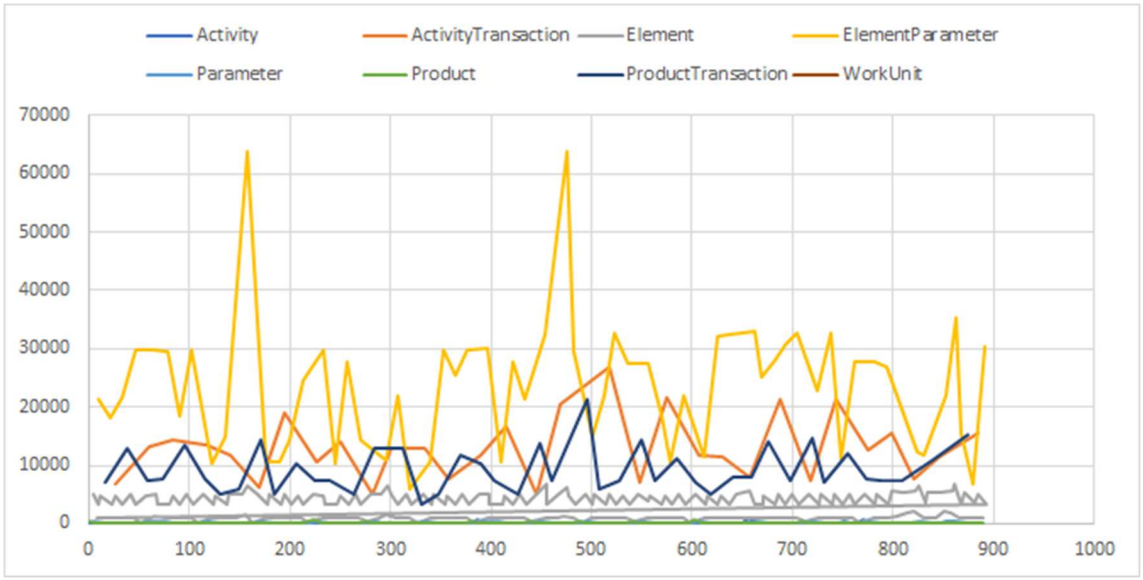
Genereeritud elementide hulk ühes tsüklis: 49990-50000. Kokku tehtud päringute arv 893. Täielik ülevaade Lisas 2 tabelites 40 ja 41. Selgelt näha kaks tekkinud ekstreemumit ElementParameter'ite salvestamisel. Tekkinud ekstreemumite põhjus teadmata.

Element	3299648
ElementParameter	13330137
WorkUnit	2769278
ActivityTransaction	2700174
ProductTransaction	2700174

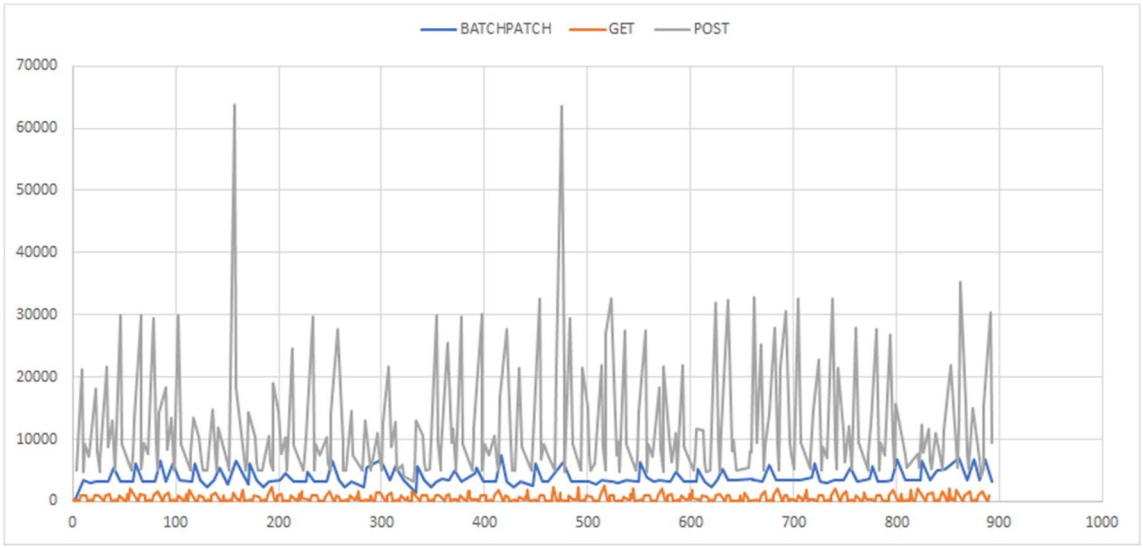
Tabel 13. Simulatsioonis tekkinud kirjete arvud(Modulaarne monoliit 50k)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	min	max	avg	min	max	avg	min	max	avg	min	max
46645	0	612396	5185209	11	62717240	4375	196	63807	1429	0.001	10225

Tabel 14. Simulatsiooni kokkuvõte (Modulaarne monoliit 50k)



Joonis 15. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(Modulaarne monoliit 50k-1)



Joonis 16. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(Modulaarne monoliit 50k)

MODULAARNE MONOLIIT 100k-1

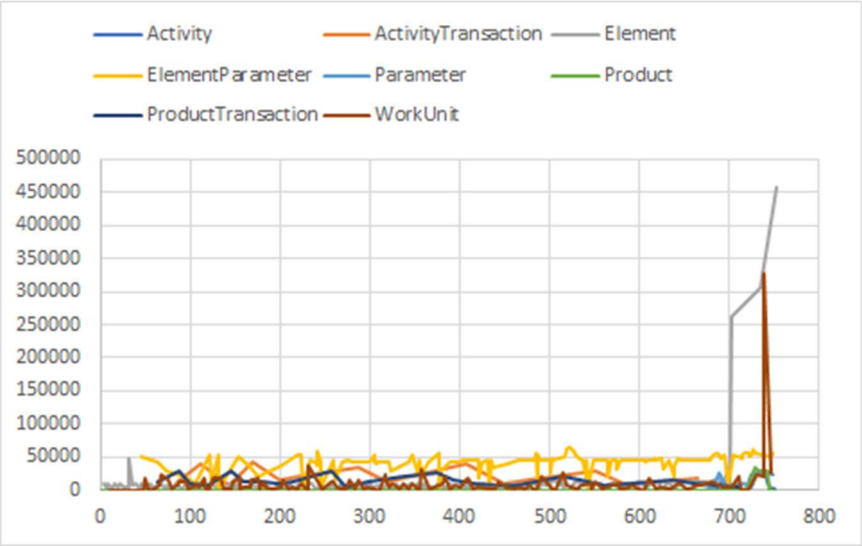
Genereeritud elementide hulk ühes tsüklis: 99990-100000. Kokku tehtud päringute arv 752. Täielik ülevaade Lisas 2 tabelites 42 ja 43. Simulatsiooni töö lõppes enneaegselt. Selgelt tuleb välja, mis hetkest süsteem tööga hakkama ei saa. Elementide salvestamise ja WorkUnit'ite pärimise aja meeletu tõusu põhjus teadmata.

Element	5099744
ElementParameter	14342224
WorkUnit	2269415
ActivityTransaction	2322832
ProductTransaction	2269415

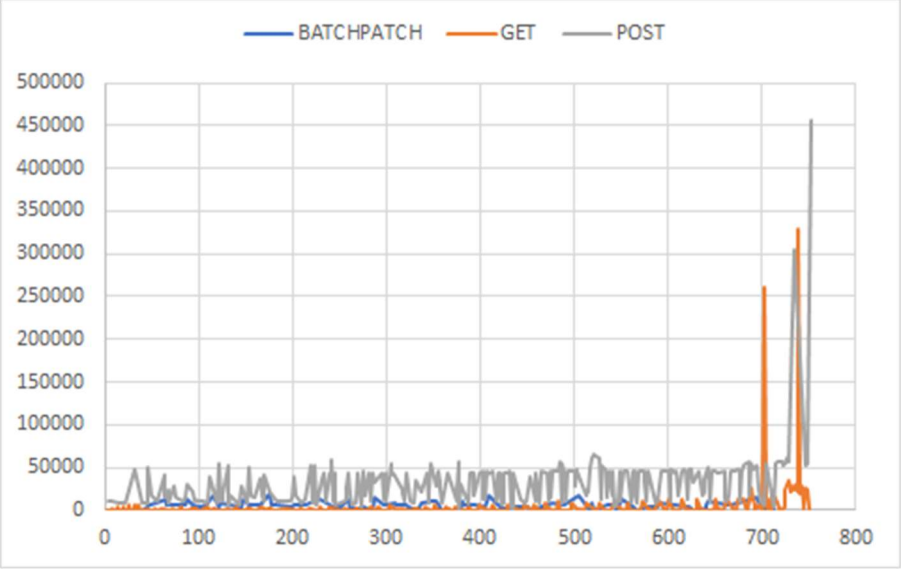
Tabel 15. Simulatsioonis tekkinud kirjete arvud(Modulaarne monoliit 100k-1)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	min	max	avg	min	max	avg	min	max	avg	min	max
75855	0	1189560	7866325	11	102698716	15337	196	456548	1170	0.0004	10447

Tabel 16. Simulatsiooni kokkuvõte (Modulaarne monoliit 100k-1)



Joonis 17. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa(Modulaarne monoliit 100k-1)



Joonis 18. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa(Modulaarne monoliit 100k-1)

MODULAARNE MONOLIIT 100k-2

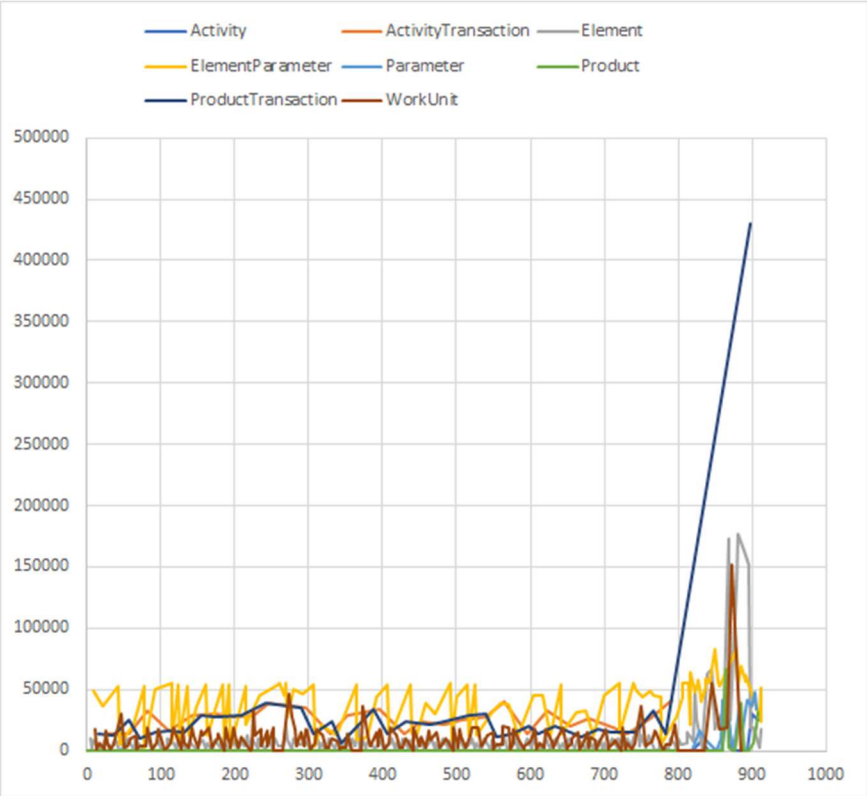
Genereeritud elementide hulk ühes tsüklis: 99990-100000. Kokku tehtud päringute arv 913. Täielik ülevaade Lisas 2 tabelites 44 ja 45. Selgelt tuleb välja, et enne simulatsiooni lõppu kasvab hetkeks kasutava aja hulk kolmekordseks.

Element	7199606
ElementParameter	28112887
WorkUnit	4930428
ProductTransaction	5060330
ActivityTransaction	4535822

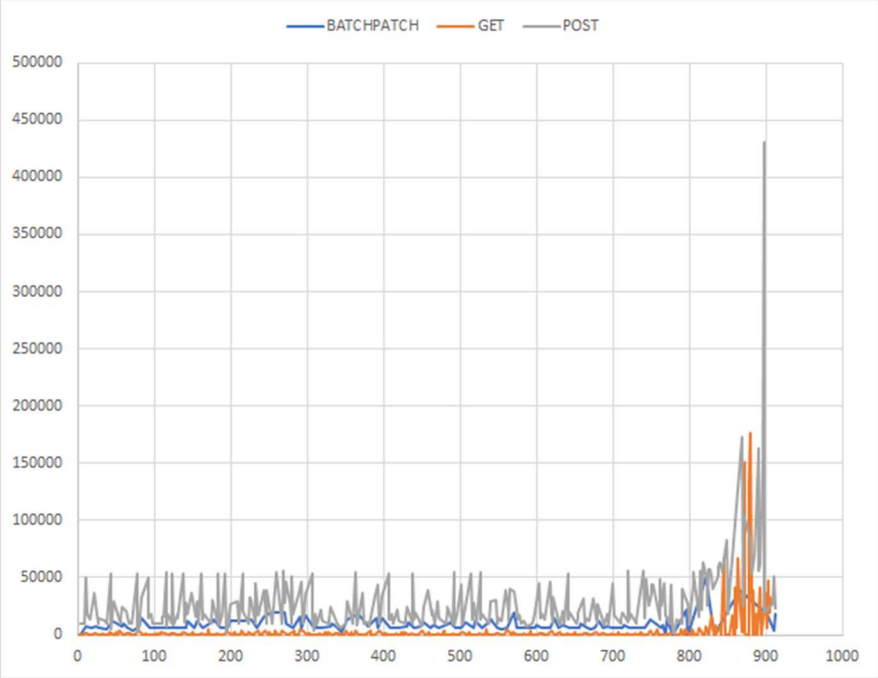
Tabel 17. Simulatsioonis tekkinud kirjete arvud(Modulaarne monoliit 100k-2)

päringu tulemuses ridu			päringu suurus baitides			päringu aeg millisekundites			suurus/aeg		
avg	max	min	avg	max	min	avg	max	min	avg	max	min
651	38092	197	831	841	831	10	10	10	4	4	0.02

Tabel 18. Simulatsiooni kokkuvõte (Modulaarne monoliit 100k-2)



Joonis 19. Päringu kestvuse [ms] sõltuvus päringu indeksist olemi kaupa.(Modulaarne monoliit 100k-2)



Joonis 20. Päringu kestvuse[ms] sõltuvus päringu indeksist päringu tüübi kaupa.(Modulaarne monoliit 100k-2)

5.2 Analüüs

Autor arvab, et süsteemide ehitamisele sätestatud piirangud said kõik tagatud. Peamiseks murekohaks pidas autor suure andmemahu tekitamist. Arvestades simulatsioonides tekitatud kirjete arvu (kahel korral ühe olemi kohta üle 28 miljoni), võib lugeda selle piirangu tagatuks. Lisaks on loodud süsteemide andmebaasid loodud modulaarselt ja nende vahel on tagatud ka pidev andmevoog.

Simulatsioonide läbimisel lõppes enneaegselt üks sünonüümidel ning üks monoliitsel arhitektuuril põhinev simulatsioon. Eriti huvitav on olukord sünonüümidel põhineva rakendusega, kus ühel korral süsteem vastu ei pidanud, kuid teisel korral läbiti simulatsioon stabiilselt. Üheks võimalikuks põhjuseks süsteemide koormuse kasvamisel on andmebaasis toimuvad automaatsed korrastustööd.

Monoliitsel arhitektuuril põhineva rakenduse puhul on erinevates simulatsioonides päringute aja suur kasv ajaliselt lähestikku. Võib teha järelduse, et piisavalt suure andmemahu juures võib monoliitsetel süsteemidel sissetulevate päringutega tegelemiseks arvutusjõud otsa saada ja ei suudeta enam rakenduse tööd tagada.

Simulatsioonides kulub kõige suurem aeg ElementParameter'ite, ActivityTransaction'ite ja ProductTransaction'itega seotud päringute tegemiseks. ElementParameter'itele kulub rohkem

aega, sest neid tekitatakse iga elemendi kohta kuni kuus tükki. Autor arvab, et ElementParameter'itega seotud ekstreemumite juures tekitati süsteemi keskmisest rohkem parameetreid ning sellega kaasnevalt pikenes ka päringutele kuluv aeg. Küll aga ei oska autor selgitada, miks ElementParameter'itega seotud ekstreemumid keskmiselt kulunud ajaga võrreldes nii suured on. ActivityTransaction'ite ja ProductTransaction'itega seotud keskmisest suuremad ajakulud on põhjendatavad, sest simulatsiooni käigus läbitakse ostu- ja tootmistsükleid märgatavalt harvemini ja päringute mahud on proportsionaalselt suuremad.

Hinnates rakenduste tööd stabiilsetel hetkedel tuleb välja, et loodud süsteemide kiirused on praktiliselt samad. Üheks põhjuseks selle juures on päringutega kaasnevate andmete simulatsioonist serveri poolsesse rakendusse üle kandmisel tekkiv viit, mis oli selgelt eristatav just suuremahuliste päringute korral. Kahjuks puuduvad andmed päringute liigutamisele kulunud ajale ja seega ei saa seda täpsemalt analüüsida.
Loodud lahendusest tooks autor välja sünonüümide haldamiseks vajaliku tehnilise konfiguratsiooni. Sünonüümide kasutamine vajab rakenduses põhjalikku seadistust ning selle tegemine ei ole triviaalne. Võib arvata, et sünonüümide kasutamisel võib süsteemi skaleerimisel tekkida probleeme dünaamilisusega.
Mikroteenustel põhineval rakendusel kasutati staatilist gateway'd, mis kõlbab ainult testimise eesmärgil loodud rakendustesse. Lisaks tuleb arvesse võtta, et rakendused töötasid kõik samal serveril, mille tõttu puudusid mikroteenuste suhtlemisel tekkivad ülekande kiirusest tingitud ajakaod.
Monoliitsel arhitektuuril põhineva rakenduse juures leidis kinnitust asjaolu, et ühe rakenduse võimekus andmete töötlemisel on piiratud. Tuleb märkida, et autor oleks oodanud alammoodulitega seotud andmete töötlemisel suuremaid päringu aegu.
Simulatsioonide tulemustest võib tuua ka teatud paralleele pärismaailmasse. Olukororras, kus monoliitsel arhitektuuril saab andmete käitlemiseks vajaminev ressurss otsa ning süsteem seiskub, ei ole seda rakendust võimalik enam kasutada.

Simulatsioonide tulemusi arvesse võttes arvab autor, et autori töökohas arendatava süsteemi probleemide lahenduseks oleks parim lahendus liikuda mikroteenustel põhinevale arhitektuurile. Küll aga tuleks selle lahenduse sobivust ja võimalikku üleminekut lähemalt uurida.

5.3 Edasiarendamise võimalused

Töö tulemusena tekkis selge ülevaade erinevaid arhitektuurilisi lahendusi kasutavate rakenduse võimekusest, kuid antud ülevaade on väga laiapõhine ja ei võta arvesse mitmeid tehnilisi aspekte.
Kuna tehtud analüüsi tulemusena selgub, et autori töökohas tuleks süsteem viia üle mikroteenustel põhinevale süsteemile, siis peab autor kõige tähtsamaks selle lahenduse lähemat uurimist. Autor kasutas teenuste vahelise suhtluse tagamiseks staatilist lahendust, reaalsuses ei ole see lahendus jätkusuutlik ja tuleks kindlasti välja vahetada. Lisaks tuleks uurida süsteemi tööd, kui teenused asuvad erinevatel serveritel ja erinevatel instantsidel. Saadud tulemustes ei tule välja teenuste ühendamisel kasutatavate päringute ajalist kulu, kuid see võib mängida suurt rolli kogu süsteemi töös.

Lisaks tuleks välja uurida erinevate lahenduste skaleeritavus. Selle all peab autor silmas süsteemi võimekuse tõstmist samaaegselt töötavate rakenduste arvu tõstmise näol. Tuleks välja selgitada, mis on töös valitud erinevate lahenduste skaleerimise võimalused ja kuidas on võimalik tagada see dünaamiliselt.

Samuti oleks oluline analüüsida simulatsiooni käigus riistvara(antud töö puhul virtuaalserveri) koormust.

Kuna loodud simulatsioonid on oma olemuselt lihtsa ülesehitusega, ei tohiks unustada, et päriselt on süsteemide sisemine keerukus märkimisväärselt suurem. Siinkohal peab autor oluliseks ettevõttes arendatava rakenduse põhjalikku analüüsi. Vajalik oleks kaardistada kõik vajadused ja sellele tuginedes, arvestades tehtud antud töö tulemusi, saab võtta vastu otsuse, milline lahendus oleks autori töökohas kõige parem.

6 Kokkuvõte

Tarkvaraarenduses on pikka aega olnud peamiseks viisiks rakenduste loomisel arendada need ühise tükina. Viimastel aastatel on aga hakatud tarkvaralisi süsteeme ehitama väiksemate funktsionaalsete tükkide kokkupanemisega.

Käesoleva töö eesmärgiks oli välja selgitada ühes ettevõttes arendatavale monoliitsele süsteemile võimalikud arhitektuursed alternatiivid ja analüüsida nende jõudlust suurte andmehulkade juures.

Autor valis 3 arhitektuurset võimalust ning nende põhjal realiseeriti töötavad süsteemid. Tulemuste saamiseks lõi autor süsteemidele simulatsioonid, mis jäljendasid päriselu. Tulemusteks on kogutud andmed üheksast simulatsioonist töös loodud süsteemide võimekuse kohta.

Tulemustes kajastuvad süsteemide võimekused ja kitsaskohad erinevate koormuste juures. Tulemuste analüüsist selgus, et monoliitsetel ja sünonüümidel ehitatud rakendustel võib tekkida probleeme suurte andmete töötlemisel.

Analüüsi käigus jõudis autor järelduseni, et parimaks lahenduseks oleks viia ettevõttes arendatav süsteem üle mikroteenustel põhinevale arhitektuurile, sest nii on võimalik tagada süsteemi töökindlus ka suurte mahtude juures. Autori esialgne arvamus, et ettevõtte süsteem tuleks jagada tükkideks, leidis töös kinnitust. Antud töö tulemusi kasutatakse ettevõttes ühe sisendina, et otsustada, millise arhitektuuriga jätkata.

7 Viited

1. Amaral, M.,Polo, J., Carrera,D, Mohomed, I.,Unuvar, M., Steinder, M., Performance Evaluation of Microservices Architectures Using Containers. IEEE, 2015, [*online*] https://ieeexplore.ieee.org/
2. Bass, L., Clements, P., Kazman, R., Software architecture in practice, 4th edition. Addison-Wesley Professional, 2021, [*online*] https://oreilly.com
3. Containers- Deep Drive. [WWW] https://aws.amazon.com/getting-started/deep-dive-containers/ (17.05.2021)
4. Cybernetica AS, *Andmekaitse ja Infoturbe Leksikon* [WWW] https://akit.cyber.ee/#showlist (17.05.2021)
5. Farmer,R., Jain, R., Wu, D., Cloud Foundry for Developers. Packt Publishing, 2021, [*online*] https://oreilly.com
6. Guerrero,S. Microservices in SAP HANA XSA: A Guide to REST APIs Using Node.js. Texas: Apress, 2020.
7. Haun,J., Hickman, C., Loden, D., Wells,R., Implementing SAP HANA. Boston: Galileo Press, 2013.
8. Hernandez, M. J., Database Design for Mere Mortals®: A Hands-on Guide to Relational Database Design, Third Edition. Addison-Wesley Professional, 2020, [*online*] https://oreilly.com
9. Hill, R., Rezai, M., Shadija, D., Microservices: Granularity vs. Performance. UCC '17 Companion: Companion Proceedings of the10th International Conference on Utility and Cloud Computing, lk 215-220, 2017, [*online*] https://dl.acm.org
10. Martin,F., Microservices [WWW] https://martinfowler.com/articles/microservices.html (17.05.2021)
11. SAP HANA Deployment Infrastructure. [WWW] https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/3ef0ee9da11440e4b01708455b8497a9.html (17.05.2021)
12. SAP HANA Developer Guide for XS Advanced Model [WWW] https://help.sap.com/viewer/4505d0bdaf4948449b7f7379d24d0f0d/2.0.03/en-US (17.05.2021)

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Kristjan Pint

Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "
Modulaarsete andmemudelite simuleerimine SAP HANA andmebaasi haldussüsteemis
", mille juhendajad on Jakob Jõgi ja Toomas Klementi

1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

18.05.2021

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2: TABELID

Olemi nimi	semantika
Element	ehitise 3D mudeli üks objekt. Kasutusel planeerimise moodulis
Parameter	Võimalik omadus, mis kirjeldab elementi. Kasutusel planeerimise moodulis
ElementParameter	Elemendi ja parameetri seos, väärtustab elemendi mingile parameetrile kindla väärtuse. Kasutusel planeerimise moodulis
WorkUnit	Tööühik. Elemendi ja Tegevuse vaheline seos vajalike väärtustega. Kasutusel planeerimis-, tootmis- ja ostumoodulis
Activity	Tegevus, mida tuleb elemendi paigalduseks teha. Kasutusel planeerimis-, osu- ja tootmismoodulis
Product	Toode, mida on tegevuse tegemiseks vaja. Kasutusel planeerimis- ja ostumoodulis
ActivityTransaction	Töö tegemise ühik koos vajalike väärtustega. Kasutusel tootmismoodulis
ProductTransaction	Toote ostmist tähistav olem koos vajalike väärtustega

Tabel 19. Olemite loetelu

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
NAME	10 karakteri pikkune juhuslik genereeritud väärtus
TYPE	“ARCH”,”STR” või “MEP”
STATUS	“NOTSAVED”, “REMOVED” või “SAVED”

Tabel 20. Olem Element

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
NAME	“Count”, “Width”, “Height”, “Thickness”, “Color”, “Type”, “Level”, “Absorptance”, “Durability” või “Heat Capacity”
UNIT	“PC”, “mm”, “rgb”, “integer”, “au”, “kg” või “c”

Tabel 21. Olem Parameter

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
ELEMENT.Id (FK)	Võõrvõti. Ühe olemi Element kirje identifikaator
PARAMETER.Id (FK)	Võõrvõti. Olemi Parameter kirje Identifikaator
UNIT	Olemi Parameter ühik. Kasutatakse olemi WorkUnit väärtusena. Saadakse Parameetri väärtusest.
UNITVALUE	Olemi Parameter ühiku väärtus. Kasutatakse olemi WorkUnit väärtuse arvutamisel.

Tabel 22. Olem ElementParameter

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
NAME	“Install wall”, “Pour concrete”, “install floor”, “install ligths”, “assemble furniture”, “install roof”, “brick layering”, “install window” või “insulate wall”
UNIT	“D/M3”, “H/M2”, “H/PC”
UNITVALUE	“1”,”3”,”5”, “10”

Tabel 23. Olem Activity

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
NAME	“Install wall”, “Pour concrete”, “install floor”, “install ligths”, “assemble furniture”, “install roof”, “brick layering”, “install window” või “insulate wall”
TYPE	“Purchase”, “Make”
PRICE	“5”, “10”, “50”, “75”, “100” või “1000”
UNIT	“M3”, “M2” või “PC”
UNITVALUE	“0.1”, “1”,”2” või “5”

Tabel 24. Olem Product

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
ELEMENT.Id (FK)	Võõrvõti. Olemi Element identifikaator
ACTIVITY.Id (FK)	Võõrvõti. Olemi Element identifikaator
UNIT	Elemendiga seotud parameetri vällja UNIT väärtus
UNITVALUE	Juhuslikult valitud Activity välja UNITVALUE väärtus
STATUS	“CONFIRMED”, “READY” või “FINISHED”
DURATION	Arvutatakse väärtustatava Parameetri ja Activity väljadest UNITVALUE
STARTTIME	Juhuslik kuupäev 01/01/2021 ja 01/01/2023 vahel

Tabel 25. Olem WorkUnit

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
WORKUNIT_Id	Ühe WorkUniti identifikaator, kuid ei ole võõrvõti
PRODUCT_Id	Ühe Producti identifikaator, kuid ei ole võõrvõti
STATUS	“PURCHASED”
PRICE	Product’i ostuhind
ARRIVALTIME	juhuslik kuupäev 01/01/2020 ja WorkUnitile välja STARTTIME vahel

Tabel 26. Olem ProductTransaction

Välja nimi	Väärtused
Id (PK)	Andmete salvestamisel tekkiv unikaalne identifikaator
WORKUNIT_Id	Ühe WorkUniti identifikaator, kuid ei ole võõrvõti
ACTIVITY_Id	Ühe Actvity identifikaator, kuid ei ole võõrvõti
STATUS	“STARTED”
TYPE	“ASSEMBLY”
ENDTIME	juhuslik kuupäev WorkUnit’i välja STARTTIME ja 01/01/2023 vahel

Tabel 27. Olem AcitvityTransaction

Sünonüümid 50k

olemi nimi		activity	activity transaction	element	element parameter	para-meter	product	product transaction	work unit
päringu tulemuses ridu	avg	10	91235	48256	218454	10	10	63465	64429
	min	10	19949	0	49991	10	10	19913	0
	max	10	196127	99995	569420	10	10	124134	196127
päringu suurus baitides	avg	831	10417898	3295914	19068689	420	902	7231895	9253994
	min	831	2277910	11	4369764	420	902	2268950	11
	max	841	22395608	6933228	49777741	420	902	14145192	28351774
päringu aeg milli-sekundites	avg	209	13852	3056	24054	227	214	9699	4839
	min	197	3450	200	5915	203	197	3336	226
	max	740	28798	9775	61774	291	699	18258	18776
suurus/ aeg	avg	4	742	1660	784	2	4	737	3298
	min	1	660	0.02	657	1	1	669	0.03
	max	4	778	4643	819	2	5	779	9587
kogus		140	33	230	69	89	139	49	239

Tabel 28 Simulatsiooni kokkuvõte olemite kaupa(Sünonüümid 50k)

päringu tüüp		batchpatch	get	post
päringu tulemuses ridu	avg	63948	18426	95722
	min	10	0	19913
	max	196127	196127	569420
päringu suurus baitides	avg	7327200	2110443	9322334
	min	831	11	2268950
	max	28351774	27763393	49777741
päringu aeg millisekundites	avg	4330	521	12293
	min	227	197	3336
	max	12546	2896	61774
suurus/aeg	avg	1570	1533	732
	min	4	0.02	355
	max	2330	9587	819
kogus		152	544	292

Tabel 29. Simulatsiooni kokkuvõte päringu tüübi kaupa(Sünonüümid 50k)

Sünonüümid 100k-1

olemi nimi		activity	activity transaction	element	element parameter	parameter	product	product transaction	work unit
päringu tulemuses ridu	avg	10	132517	192900	91191	10	10	108597	93202
	min	10	4398	0	31	10	10	4398	0
	max	10	286494	1365780	500000	10	10	286494	286494
päringu suurus baitides	avg	831	15316187	13341134	7969827	420	902	12624670	13496627
	min	831	511025	11	42	420	902	511341	11
	max	841	33000864	94694330	43710648	420	902	33305746	41987404
päringu aeg millisekundites	avg	6002	20649	26698.0	58689	9249	3117	38399	20149
	min	196	2734	199	2744	202	196	1304	217
	max	96702	42165	334270	658580	79518	77969	223965	337376
suurus/ aeg	avg	3	681	1838	197	1	3	629	2697
	min	0.009	187	0.0004	0.0003	0.005	0.01	23	0.0003
	max	4	795	6131	838	2	5	798	10287
kogus		107	18	146	165	77	106	27	143

Tabel 30. Simulatsiooni kokkuvõte olemite kaupa(Sünonüümid 100k-1)

päringu tüüp		batchpatch	get	post
päringu tulemuses ridu	avg	80819	65546	95325
	min	0	0	30
	max	286494	1365780	500000
päringu suurus baitides	avg	9997647	5565213	9073324
	min	11	11	41
	max	41700910	94694330	43710648
päringu aeg millisekundites	avg	9822	10916	48075
	min	206	196	1304
	max	111752	334270	658580
suurus/aeg	avg	1116	1252	377
	min	0.0004	0.0003	0.0003
	max	2287	10287	838
kogus		85	407	297

Tabel 31. Simulatsiooni kokkuvõte päringu tüübi kaupa(Sünonüümid 100k-1)

Sünonüümid 110k-2

olemi nimi		activity	activity transaction	element	Element parameter	parameter	product	product transaction	workunit
päringu tulemuses ridu	avg	10	171678	100432	301336	10	10	122205	122358
	min	10	63939	0	31	10	10	40116	0
	max	10	299128	299982	500000	10	10	271326	299128
päringu suurus baitides	avg	10	19946956	6877406	26389729	420	902	14206649	17713948
	min	831	7428902	6877406	42	420	902	4663491	11
	max	841	34754089	2079858 5	44208274	420	902	31542189	43539531
päringu aeg millisekundites	avg	214	25589	5739	33426	237	204	18754	9041
	min	197	11072	199	2753	205	196	6393	217
	max	958	43815	18391	56754	389	318	40241	36586
suurus/ aeg	avg	4	770	2044	757	2	4	755	3814
	min	0.9	671	0.05	0.002	1	3	434	0.001
	max	4	803	6083	837	2	5	802	10481
kogus		138	33	229	95	89	137	47	235

Tabel 32. Simulatsiooni kokkuvõte olemite kaupa(Sünonüümid 100k-2)

päringu tüüp		batchpatch	get	post
päringu tulemuses ridu	avg	126233	36689	169337
	min	0	0	30
	max	299982	299982	500000
päringu suurus baitides	avg	14340907	4133413	16567286
	min	11	11	41
	max	43539531	42642147	44208274
päringu aeg millisekundites	avg	8266	756	21588
	min	207	196	2753
	max	20597	7723	56754
suurus/aeg	avg	1634	1890	746
	min	0.05	0.01	0.001
	max	2378	10481	837
kogus		149	540	314

Tabel 33. Simulatsiooni kokkuvõte päringu tüübi kaupa(Sünonüümid 100k-2)

Mikroteenused 50k

olemi nimi		activity	activity transaction	element	element parameter	parameter	product	product transaction	workunit
päringu tulemuses ridu	avg	10	45531	60086	217800	10	10	55545	422631
	min	10	43389	0	29	10	10	31825	0
	max	10	47673	811	612040	10	10	94708	1999476
päringu suurus baitides	avg	811	5034400	5076298	22077578	410	0	6174908	59894378
	min	811	4824405	11	40	410	1092	3540406	11
	max	831	5244395	38396062	62070842	410	1092	10535818	286736668
päringu aeg milli-sekundites	avg	265	8463	3496	26713	248	213	12687	12113
	min	197	8286	198	5694	199	213	5220	265
	max	1012	8640	23827	141890	632	213	55317	53135
suurus/ aeg	avg	3	595	2188	886	2	5	627	4020
	min	0.8	582	0.04	0.001	0.6	5	190	0.04
	max	4	607	7194	967	2	5	742	8184
kogus		142	2	218	58	89	141	12	159

Tabel 34. Simulatsiooni kokkuvõte olemite kaupa(Mikroteenused 50k)

päringu tüüp		batchpatch	get	post
päringu tulemuses ridu	avg	53573	128061	97789
	min	10	0	29
	max	99993	1999476	612040
päringu suurus baitides	avg	4995402	17487396	10209205
	min	811	11	40
	max	13027523	286736668	62070842
päringu aeg millisekundites	avg	3844	2822	13298
	min	238	196	4077
	max	13489	53135	141890
suurus/aeg	avg	1317	1688	785
	min	3	0.04	0.001
	max	2064	8184	967
kogus		72	548	201

Tabel 35. Simulatsiooni kokkuvõte päringu tüübi kaupa(Mikroteenused 50k)

Mikroteenused 100k-1

olemi nimi		activity	activity transaction	element	element parameter	parameter	product	product transaction	workunit
päringu tulemuses ridu	avg	10	169612	98795	312795	10	10	129053	120116
	min	10	39946	0	21511	10	10	39946	0
	max	10	316553	299984	500000	10	10	260895	316553
päringu suurus baitides	avg	831	169612	8450445	32661150	420	1102	14732872	17282847
	min	831	39946	11	2246330	420	1102	4555644	11
	max	841	316553	25898706	52211147	420	1102	29754697	45758770
päringu aeg milli-sekundites	avg	225	169612	5851	34589	254	266	19940	9659
	min	196	39946	198	2707	203	198	6739	235
	max	1205	316553	15633	55945	1223	6178	39089	33175
suurus/aeg	avg	4	169612	2417	914	2	5	731	2885
	min	0.7	39946	0.02	591	0.3	0.2	628	0.001
	max	4	316553	6694	991	2	6	763	7918
kogus		151	35	250	97	97	147	46	249

Tabel 36. Simulatsiooni kokkuvõte olemite kaupa(Mikroteenused 100k-1)

päringu tüüp		batchpatch	get	post
päringu tulemuses ridu	avg	124339	35809	169691
	min	10	0	30
	max	316553	316553	500000
päringu suurus baitides	avg	14960618	4259017	18379753
	min	831	11	41
	max	45758770	44809111	52211147
päringu aeg millisekundites	avg	8674	920	23658
	min	224	196	2707
	max	24189	11485	113483
suurus/aeg	avg	1606	1629	792
	min	4	0.02	0.001
	max	2165	7918	991
kogus		155	583	334

Tabel 37. Simulatsiooni kokkuvõte päringu tüübi kaupa(Mikroteenused 100k-1)

Mikroteenused 100k-2

olemi nimi		activity	activity transaction	element	element parameter	parameter	product	product transaction	workunit
päringu tulemuses ridu	avg	217	39181	5642	31598	233	210	19603	9392
	max	663	86001	17288	56671	373	344	30236	36939
	min	196	13707	196	2871	203	198	6818	231
päringu suurus baitides	avg	831	18504754	8064953	29776234	420	1102	14661669	17310759
	max	841	33445456	17266903	52210873	420	1102	23008111	41899742
	min	831	7390144	11	2234249	420	1102	4647287	4647287
päringu aeg millisekundites	avg	10	159820	94368	285168	10	10	126466	119573
	max	10	287862	199998	500000	10	10	198004	287862
	min	10	63605	0	21402	10	10	39993	0
suurus/aeg	avg	4	495	2402	901	2	5	743	2983
	max	4	600	6718	992	2	6	772	7871
	min	1	389	0.03	545	1	3	682	0.03
kogus		135	35	231	95	89	135	45	238

Tabel 38. Simulatsiooni kokkuvõte olemite kaupa(Mikroteenused 100k-2)

päringu tüüp		batcpatch	post	get
päringu tulemuses ridu	avg	8621	22372	893
	max	21047	86001	18777
	min	232	2871	196
päringu suurus baitides	avg	14755320	17731282	893
	max	41899742	52210873	41036156
	min	831	2234249	11
päringu aeg millisekundites	avg	122413	162348	4236153
	max	287862	500000	287862
	min	10	21402	0
suurus/aeg	avg	1630	799	1685
	max	2167	992	7871
	min	4	389	0.03
kogus		151	317	536

Tabel 39. Simulatsiooni kokkuvõte päringu tüübi kaupa(Mikroteenused 100k-2)

Modulaarne monoliit 50k

olemi nimi		activity	activity transaction	element	element parameter	parameter	product	product transaction	workunit
päringu tulemuses ridu	avg	10	87102	49756	215003	10	10	61368	59624
	min	10	31847	0	29	10	10	19872	0
	max	10	187592	99999	612396	10	10	148180	187592
päringu suurus baitides	avg	811	9858900	4203535	22019793	410	1092	6956180	8450722
	min	811	3604566	11	40	410	1092	2252707	11
	max	831	21232488	8533333	62717240	410	1092	16796788	26179163
päringu aeg milli-sekundites	avg	219	12974	3009	23607	234	217	9250	4433
	min	196	5015	200	5948	200	199	3318	223
	max	864	26958	6896	63807	307	651	21409	18377
suurus/aeg	avg	4	753	2116	928	2	5	744	3424
	min	0.9	690	0.02	0.001	1	2	679	0
	max	4	793	6179	991	2	5	785	10225
kogus		128	31	209	62	82	127	44	210

Tabel 40. Simulatsiooni kokkuvõte olemite kaupa(Modulaarne monoliit 50k)

päringu tüüp		batchpatch	get	post
päringu tulemuses ridu	avg	58854	18444	93937
	min	10	0	29
	max	115956	187592	612396
päringu suurus baitides	avg	6810720	2182279	10060761
	min	811	11	40
	max	16530041	26179163	62717240
päringu aeg millisekundites	avg	3938	513	11891
	min	228	196	3318
	max	7460	2594	63807
suurus/aeg	avg	1667	1693	814
	min	4	0.02	0.001
	max	2271	10225	991
kogus		130	499	264

Tabel 41. Simulatsiooni kokkuvõte päringu tüübi kaupa(Modulaarne monoliit 50k)

Modulaarne monoliit 100k-1

olemi nimi		activity	activity transaction	element	element parameter	parameter	product	product transaction	workunit
päringu tulemuses ridu	avg	9	138083	192402	72739	9	9	90777	82777
	min	0	40107	0	29	0	0	6674	0
	max	10	291417	1189560	500000	10	10	206473	291417
päringu suurus baitides	avg	771	15813968	16586743	7593001	384	1022	10412752	11927731
	min	11	4579930	11	40	11	11	769879	11
	max	841	33276659	102698716	52209806	420	1102	23611015	42125666
päringu aeg milli-sekundites	avg	1493	21375	13616	42705	2918	1508	14356	9825
	min	198	6439	207	1641	203	196	4551	213
	max	30135	41990	456548	65134	27354	34508	29842	328086
suurus/aeg	avg	3	718	2609	226	1	4	699	3026
	min	0.03	540	0.0007	0.0006	0.02	0.03	156	0.0004
	max	4	792	7235	985	2	6	791	10447
kogus		96	16	153	162	68	95	25	137

Tabel 42. Simulatsiooni kokkuvõte olemite kaupa(Modulaarne monoliit 100k-1)

päringu tüüp		post	batchpatch	get
päringu tulemuses ridu	avg	82777	68357	72454
	min	0	0	0
	max	291417	291417	1189560
päringu suurus baitides	avg	11927731	8829906	6950756
	min	11	11	11
	max	42125666	42125666	102698716
päringu aeg millisekundites	avg	9825	6120	4147
	min	213	207	196
	max	328086	19225	328086
suurus/aeg	avg	3026	1119	1728
	min	0.0004	0.0007	0.0004
	max	10447	2398	10447
kogus		137	86	380

Tabel 43. Simulatsiooni kokkuvõte päringu tüübi kaupa(Modulaarne monoliit 100k-1)

Modulaarne monoliit 100k-2

olemi nimi		activity	activity transaction	element	element parameter	parameter	product	product transaction	workunit
päringu tulemuses ridu	avg	1285	25525	11123	39156	3825	1527	32024	10486
	max	38092	40846	176435	83238	47507	66692	430464	151190
	min	197	13366	200	2770	203	199	6481	199
päringu suurus baitides	avg	831	19479223	9661646	23781182	420	1102	15921098	17294462
	max	841	32473744	54768928	52210196	420	1102	30953092	40681832
	min	831	9996100	11	40	420	1102	4635915	11
päringu aeg milli-sekundites	avg	10	167993	112724	227757	10	10	137088	119436
	max	10	279491	634394	500000	10	10	266151	279491
	min	10	86779	0	29	10	10	39861	0
suurus/aeg	avg	4	758	2321	667	2	5	740	3464
	max	4	800	7055	989	2	6	796	10862
	min	0.02	443	0.0002	0.0005	0.009	0.02	71	0.0006
kogus		127	27	209	105	89	126	35	195

Tabel 44. Simulatsiooni kokkuvõte olemite kaupa(Modulaarne monoliit 100k-2)

päringu tüüp		batchpatch	get	post
päringu tulemuses ridu	avg	9404	3248	28493
	max	49674	176435	28493
	min	207	197	28493
päringu suurus baitides	avg	14483148	4675904	16851610
	max	40681832	54768928	16851610
	min	11	11	16851610
päringu aeg millisekundites	avg	119437	41272	155452
	max	299984	634394	155452
	min	0	0	155452
suurus/aeg	avg	1576	1743	727
	max	2345	10862	727
	min	0.0002	0.0006	727
kogus		122	503	288

Tabel 45. Simulatsiooni kokkuvõte päringu tüübi kaupa(Modulaarne monoliit 100k-2)