

TALLINNA TEHNIKAÜLIKOOL
INFOTEHNOLOOGIA TEADUSKOND

Arvutiteaduse instituut
Võrgutarkvara õppetool

Logifailide töötlemise ja visualiseerimise rakendus: võrdlus, analüüs ja arendus

Bakalaureusetöö

Üliõpilane: Chris Kokka
Üliõpilaskood: 121061
Juhendaja: Ago Luberg

Tallinn
2015

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.

(kuupäev)

(allkiri)

Annotatsioon

Käesoleva töö eesmärk on leida lahendus logifailides asuvate andmete visualiseerimisele, mis aitab ekspertidel loodud graafikute põhjal teostada analüüsi enda klientidele. Eesmärgi saavutamiseks on antud töös kirjeldatud järgmised sammud: praeguse lahenduse probleemide kaardistamine, ülevaade ja võrdlus eksisteerivatest rakendustest, analüüs ja nõuete välja töötamine ning lahenduse loomine.

Töös käsitletud probleemide hulka kuuluvad: töödeldavate andmeridade arvu piirang, andmetöötluse kiirus, töödeldud andmete korduvkasutatavus, ajavahemiku muutmise paindlikkus filtreerimiseks ning graafikute salvestamine.

Kõik lõppkasutajatega koos väljatöötatud nõuded said loodud rakenduses täidetud. Olulisemad nõuded, mis täideti, on 1,85 miljoni reaga failide impordi tugi, kasutaja poolt peaprotsessi sooritamine 60 MB failiga vähem kui kahe minutiga, ühe kuu andmete välja joonistumine graafikul vähem kui 10 sekundiga ja rakenduse poolt 60 MB faili üles laadimine ja andmebaasi sisestamine keskmiselt 10 sekundiga.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 43 leheküljel, 5 peatükki, 12 joonist, 4 tabelit.

Abstract

The goal of the thesis is to find a solution to the problem of visualizing data from log files. Data visualization is necessary for the client's team members to carry out analysis based on the graphs for their clients. In order to reach the goal, this thesis has described the following steps: analysing the problems of the current solution, overview of existing solutions and their comparison, analysis and requirements elicitation, creating the application.

Problems dealt with in the thesis include the following: the limit of rows that the application can process, the speed of operation, reusage of processed data, the flexibility of changing the time period for filtering and saving the graph.

All the requirements created together with the end users were satisfied by the created application. The most important requirements that were satisfied were: importing files with up to 1,85 million rows, completing the main process by the user with a 60 MB file in under two minutes, creating the graph with one month's data in under 10 seconds and completing the main process by the application with a 60 MB file on average in 10 seconds.

The thesis is in estonian and contains 43 pages of text, 5 chapters, 12 figures, 4 tables.

Lühendite ja mõistete sõnastik

Broker

Broker

Sõnumivahetuse komponent ESB-s.

CSV

Comma Separated Values

Komaeraldusega väärtused - Porditav failivorming, kus andmebaasikirjed on üksteisest eraldatud komadega. Selles vormingus on iga rida üks kirje, mille väljad on üksteisest komadega eraldatud. Komade järel võib olla suvaline arv tühikuid ja/või tabeldusmärke (*tab character*), sest neid ignoreeritakse. [10]

Domeen

Domain

Domeeniks nimetatakse Sonic ESB komponentide kogumikku.

Drop-down

drop-down

Rippmenüü - Allapoole avanev menüü, millest normaalolekus on näha ainult üks rida. [11]

ESB

Enterprise Service Bus

ESB on tarkvara arhitektuuri mudel, integreerimaks omavahel suhtlevaid rakendusi, mida kasutatakse peamiselt teenus-orienteeritud arhitektuuris (*Service Oriented Architecture*).

ESB Container

Enterprise Service Bus Container

Teenuseid ja protsesse majutav komponent.

Ettevalmistatud lause

Prepared statement

Ettevalmistatud struktuuriga SQL lause.

FTP

File Transfer Protocol

Failiedastusprotokoll - FTP protokoll on ette nähtud failide edastamiseks ühest arvutist teise üle Interneti või muu TCP/IP võrgu. [12]

GB	<i>Gigabyte</i> Gigabait - 1 gigabait on ligikaudu 1 miljard baiti, täpsemalt $2^{30}=1\ 073\ 741\ 824$ baiti. [13]
GIF	<i>Graphics Interchange Format</i> Graafikavahetuse vorming - LZW-algoritmil põhinev bittrastri tüüpi graafikafaili vorming, mida kasutatakse eelkõige veebidokumentides. Toetab kuni 256 värvi ja erinevaid lahutusvõimeid. [14]
JPEG	<i>Joint Photographic Experts Group</i> Ühendatud Fotograafiaekspertide Grupp - JPEG tähistab nii selle ISO/CCITT töögrupi nime kui ka sama grupi poolt loodud ja väga populaarseks muutunud graafikafailide tihendamise meetodit. [15]
JVM	<i>Java Virtual Machine</i> Java Virtuaalmasin - Tarkvara, mis interpreteerib Java programmeerimiskeeles kirjutatud programme. Erinevalt teistest käsustikest toetab JVM objektorienteeritud programmeerimist otseselt, kuna selles sisalduvad käsud objektimeetodite käivitamiseks. [16]
MB	<i>Megabyte</i> Megabait - Ligikaudu 1 miljon baiti, täpsemalt 2^{20} ehk 1 048 576 baiti või 1024 kilobaiti. [17]
MF Container	<i>MainFrame Container</i> Komponent, mis võib sisaldada endas nii <i>Brokerit</i> kui ka <i>ESB containerit</i> . Iga <i>MF Container</i> konteiner on üks JVM (seletus allpool).
MVP	<i>Minimum Viable Product</i> Minimaalne Väärtuslik Toode – Tootearenduses on MVP toode, mille tootlus investeeringust võrreldes riskiga on suurim. MVP-s on ainult need funktsionaalsused, mis võimaldavad selle toote turule viia ja ei midagi muud.
Mõõdik	<i>Metric</i> Mõõdik on võti-väärtus paar, kus võti on mõõdiku nimi sõnena ja väärtus

on täisarv.

PDF

Portable Document Format

Porditav dokumendiformaat, porditav failivorming - Adobe Systems'i loodud platvormist sõltumatu vorming teksti, graafika jms esituseks. PDF-vormingu peamine eelis on selles, et dokumendi teisendamisel elektroonilisele kujule säilivad kõik paberkandjal dokumendi elemendid. [18]

Pistikprogramm

Plugin

Pistikprogramm, pistik, plugin - Tarkvaramoodul, mis lisab suuremale süsteemile teatud omaduse või teenuse. [19]

PNG

Portable Network Graphics

Porditav võrgugraafika - Bittgraafika standardvorming, mis on firma Unisys poolt patenditud ja litsentsitud ning on mõeldud asendada praegu Internetis laialdaselt kasutatavat GIF-vormingut. [20]

RAM

Random-Access Memory

Muutmälu, suvapöördusmälu - Arvuti keskne mäluseade, kuhu saab andmeid kirjutada ja kust saab neid lugeda. Suvapöördus (*random access*) tähendab seda, et igal mälupesal on oma aadress ning nii lugemiseks kui kirjutamiseks on võimalik pöörduda suvalise aadressi poole. [21]

SSD

Solid State Drive

Pooljuhtketas - Mälukiipidel realiseeritud kõvaketta analoog, mida kasutatakse kiireks andmepöörduseks või mehaaniliste kõvaketaste jaoks kahjulikus keskkonnas. [22]

VPN

Virtual Private Network

Virtuaalne Privaatvõrk - Privaatvõrk, mis kasutab avalikku telekommunikatsiooni infrastruktuuri, säilitades samal ajal privaatsuse ja turvalisuse. Turvalisuse tagamiseks kasutatakse tunneldamist ja vastavaid turvaprotseduure. [23]

Jooniste nimekiri

Joonis 1 Domeen ja selle komponendid	13
Joonis 2 Olemasoleva kliendi valimine nimekirjast või uue kliendi sisestamine.....	26
Joonis 3 Logifailide üles laadimine ja andmebaasi sisestamine.....	26
Joonis 4 Komponenti põhine filtreerimine	27
Joonis 5 Mõõdiku põhine filtreerimine	27
Joonis 6 Ajaperioodi põhjal filtreerimine.....	27
Joonis 7 Üles laetud failide nimekiri	28
Joonis 8 Graafiku printimine ja graafiku eksport	28
Joonis 9 Graafiku kuvamine	28
Joonis 10 Andmebaasiskeem.....	29
Joonis 11 Rakenduse komponentmudel	29
Joonis 12 Faili üleslaadimise ja andmebaasi sisestamise protsessi jadadiagramm	32

Tabelite nimekiri

Tabel 1 Logifailide töötlemiseks mõeldud rakendused	20
Tabel 2 Graafikute loomiseks mõeldud rakendused	22
Tabel 3 Faili mällu lugemise ja andmete andmebaasi sisestamise osakaalu võrdlus.....	33
Tabel 4 Faili mällu lugemise ja andmete andmebaasi sisestamise ajaline võrdlus	35

Sisukord

1. Sissejuhatus	11
1.1 Ülesande püstitus	12
1.2 Ülevaade tööst	12
2. Kontekst.....	13
2.1 Andmete analüüsi eesmärk	15
2.2 Praeguse lahenduse kirjeldus	15
2.3 Praeguse lahenduse probleemid.....	16
3. Analüüs	17
3.1 Nõuded.....	17
3.2 Võimalike lahenduse osade kirjeldus	19
3.3 Logifailide töötlemiseks mõeldud rakenduste võrdlus	20
3.4 Graafikute loomiseks mõeldud rakenduste võrdlus.....	22
4. Lahendus.....	24
4.1 Metoodika.....	24
4.2 Rakenduse kirjeldus.....	26
4.3 Kiiruse saavutamine	31
4.4 Tagasiside rakenduse kasutajatelt.....	35
4.5 Edasised arendused ja plaanid rakendusega	36
5. Kokkuvõte	38
Summary.....	39
Kasutatud kirjandus	40
Lisa 1	43

1. Sissejuhatus

ESB (*Enterprise Service Bus*) [28] on tarkvara arhitektuuri mudel. See on kogumik reegleid ja põhimõtteid integreerimaks suurt hulka rakendusi üle siini-põhise taristu (*infrastructure*). ESB tooted võimaldavad kasutajatel sellist arhitektuuri luua aga teevad seda erinevatel viisidel ning pakuvad erinevaid võimalusi. ESB arhitektuuri tuumkontseptsioon on integreerida erinevaid rakendusi pannes nende vahele kommunikatsioonisiini, mis võimaldab igal rakendusel siiniga suhelda. See sidestab süsteemid teineteisest lahti ning võimaldab suhtlemist ilma sõltuvusteta või teiste liidestatud süsteemide infot teadmata. [50]

Näitlikustamiseks ESB põhimõtteid on Äripäevast [7] toodud järgmine tsitaat:

Kristi Pool, EBSA AS turundus- ja müügijuht:

„ESB kasutamine annab võimaluse liita kõik kasutatavad programmid minimaalse ühenduste arvuga - lüli on vaja süsteemi ja platvormi vahele.

Antud filosoofias ei mängi rolli inimesed, vaid infosüsteemid. Ehk kui tööl on 10 inimest ja omavahel liidestatakse 200 infosüsteemi, siis tasub asi ära. Kui tööl on 400 inimest ja kasutusel on 2-3 infosüsteemi, ei ole asjal mõtet.

ESBd iseloomustab väga ilmekalt postkasti samastamine: kui me tahame saata kirja Tšiilisse, siis meil piisab vaid kirja kirjutamisest, margistamisest ning postkasti poetamisest, et see Tšiilis adressaadini jõuaks. Aga ESB on tegelikult veel võimsam - ta nimelt on võimeline sinu emakeeles kirjutatud kirja ka hispaania keelde tõlkima, vastuse hispaania keeles saama ja siis jälle sulle emakeelde tõlkima.“

ESB-d kasutatakse enamasti integratsioonilahendustes, kus on palju erinevaid infosüsteeme, mida on omavahel vaja liidestada. Selle tõttu on ka ESB-s palju, mille seisukorda tahetakse jälgida ja analüüsida. Autori ülesanne oli leida lahendus järgnevale probleemile. Sonic ESB [3] (edaspidi ESB, kuna töö sisu ei sõltu sellest konkreetsest tootest) on ESB, millega autoril on kogemust ning millega see töö on seotud. ESB genereerib iga komponendi kohta mõõdikuid. Need mõõdikud talletatakse logifailides, millest on rida rea haaval väga raske mingeid järeldusi teha, mistõttu otsibki klient lahendust, mis mõõdikute analüüsi hõlbustaks.

Kliendil on soov need andmed visualiseerida ning lasta ekspertidel loodud graafikute põhjal teha järeldusi ja soovitusi. Kirjeldatud tööd tehakse regulaarselt kord kuus või vajaduse korral. Vajadus tekib enamasti siis, kui klient soovib riistvaraplatvormi uuendada ja vajab serveri konfiguratsiooni määramisel abi või soovib olemasolevat konfiguratsiooni kontrollida.

1.1 Ülesande püstitus

Antud töös leiab autor lahenduse järgnevatele probleemidele.

- 1) Praeguse lahenduse probleemide kaardistamine
- 2) Ülevaade ja võrdlus eksisteerivatest rakendustest
- 3) Analüüs ja nõuete väljatöötamine
- 4) Lahenduse loomine

1.2 Ülevaade tööst

Teises peatükis kirjeldab autor töö konteksti, et lugejal oleks lihtsam aru saada, mis infot töödeldakse ja miks. Lisaks annab autor täpsema kirjelduse praegusest lahenduses ja esitab ka probleemid, mis sellel esinevad.

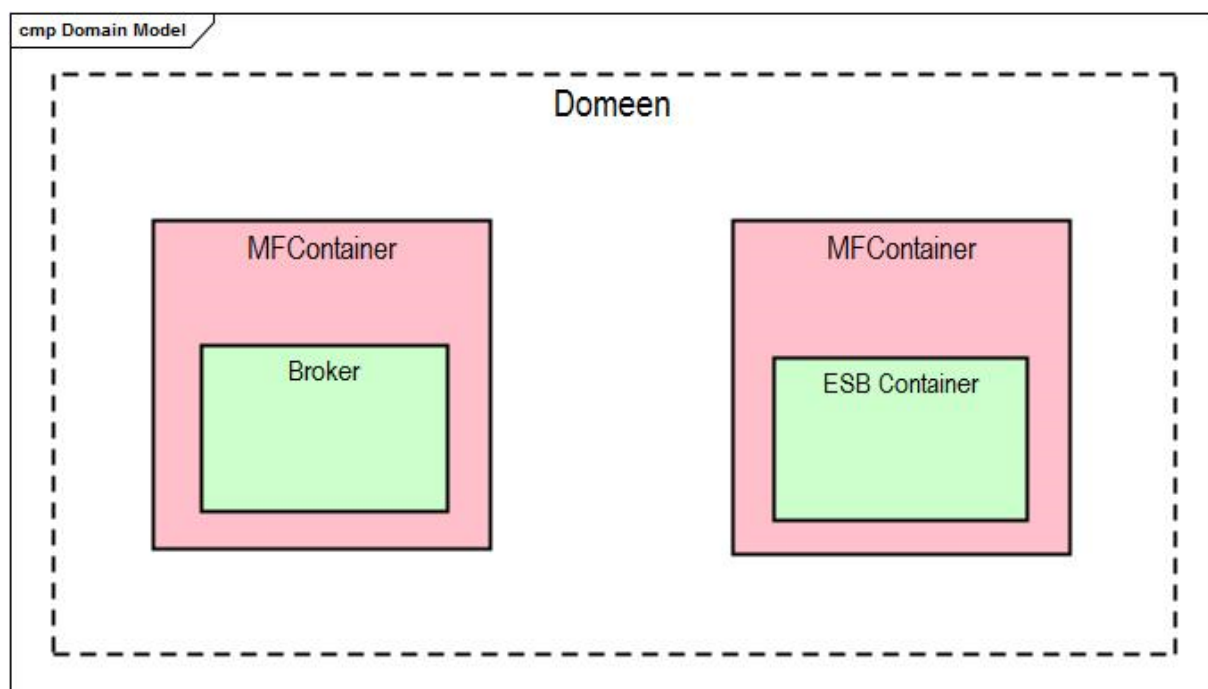
Kolmandas peatükis esitab autor nõuded, mis on väljatöötatud koos lõppkasutajatega, kes peaksid hakkama seda lahendust kasutama. Lisaks kirjeldab autor erinevate võimalike lahenduse osade kombinatsioonid. Kolmandas peatükis esitab autor ka logifailide töötlemise rakenduste ning graafikute loomiseks mõeldud rakenduste võrdlused.

Neljandas peatükis kirjeldab autor, kuidas loodi nõuetele vastav rakendus. Autor kirjeldab meetodika, mida järgis ning esitab rakenduse kirjelduse ja kasutajatelt saadud tagasiside. Autor kirjeldab ka, kuidas saavutati soovitud kiirus ning toob edasised arendused ja plaanid rakendusega.

2. Kontekst

Mõõdik – Võti-väärtus paar, kus võti on mõõdiku nimi sõnena ja väärtus on täisarv.

Erinevate ESB komponentide kogumit nimetatakse domeeniks. Logifailid, mida on vaja analüüsida, sisaldavad mõõdikuid, mida mõõdetakse erinevate ESB komponentide kohta kliendi domeenis. Nendeks komponentideks võivad olla: *MF Container*, *Broker* või *ESB Container*. *MF Container* on Sonic ESB [3] mõiste Java virtuaalmasina (JVM) kohta, mis võib majutada endas kahte ülejäänud komponenti: *Brokerit* või *ESB Containerit*. *Broker* on ESB komponent, mis tegeleb sõnumivahetuse korraldamisega ja sõnumite hoiustamisega. *ESB Container* on komponent, mis hoiab endas teenuseid ja protsesse. Igal komponendil on erinev arv ja nimetus mõõdikutel, mida saab jälgida ja logisse kirjutada. Joonisel 1 on esitatud domeen koos selle alla kuuluvate komponentidega.



Joonis 1 Domeen ja selle komponendid

Üks logi rida sisaldab endas:

- 1) Logisse kirjutamise kuupäev ja kellaaeg
- 2) Logi rea tüüp

- 3) Domeeni nimi
- 4) *MF Containeri* nimi
- 5) Komponenti nimi
- 6) Mõõdiku nimi
- 7) Mõõdiku väärtus
- 8) Ajatempel (millal mõõtmise teostati)

Logi rea näide:

```
18           Jun           2014           12:07:36           INFO
source=dmModel.ctExample:ID=AGENT;id=system.memory.MaxUsage;va
lue=959828216;timestamp=1403082365784
```

Komponentide põhiselt esinevad mõõdikud:

MF Container:

- 1) Komponenti hetke mälu kasutus.
- 2) Komponenti maksimaalne mälu kasutus.
- 3) Komponenti hetkel kasutatavate lõimede arv.

Broker:

- 1) Kohaletoimetatud sõnumite koguarv hetkel.
- 2) Vastuvõetud sõnumite koguarv hetkel.
- 3) Sekundis kohaletoimetatud andmemaht baitides.
- 4) Hetkel aktiivsete ühenduste arv.
- 5) Sekundis kohaletoimetatud sõnumite arv.
- 6) Sõnumite koguarv järjekorras (*queue*).

- 7) Sõnumite maksimaalne vanus järjekorras (*queue*).

ESB Container:

- 1) Keskmine sõnumi töötlemise aeg sekundites.
- 2) Teenuse poolt vastu võetud sõnumite arv.
- 3) Teenuse poolt vastu võetud sõnumite arv sekundis.

2.1 Andmete analüüsi eesmärk

Kogutud andmete analüüsimine võimaldab kliendil:

- 1) Ennetavalt uurida anomaaliaid kogutud andmetes.
- 2) Ennetada mälupuudust.
- 3) Soovitada muudatusi domeeni omanikule.
- 4) Jälgida süsteemi käitumist ajas.
- 5) Tagantjärei otsida vea tekkimise võimalikke põhjuseid.

2.2 Praeguse lahenduse kirjeldus

Lahendus, mida klient hetkel kasutab, on Microsoft Exceli [27] abil andmete töötlemine ja graafikute kuvamine. Ühes Exceli failis on kolm tabelit. Neist esimesele on vaja sisse lugeda kasutajal andmed. Teisel tabelil saab kasutaja filtreerida komponentide ja mõõdikute põhiselt kolmanda tabeli peal asuvat graafikut. Suurim fail, mida Excel impordib, on 175 MB - siis tuleb tabeli ridade limiit ette (1,048,576).

Praeguse süsteemi arendamist alustas klient iseseisvalt. MVP-s (*Minimum Viable Product*) [32] kerkisid aga juba esimeste testidega esile puudujäägid, mis takistasid selle lahendusega tööülesande lahendamist. Need puudujäägid on kirjeldatud järgmises peatükis.

2.3 Praeguse lahenduse probleemid

1) Limiteeritud andmeridade arv tabelis (*sheet*).

Andmekogus, mida saab korraga sisse lugeda ja töödelda, on limiteeritud. 1,048,576 rida on maksimaalne ridade arv ühes tabelis. Selline ridade arv vastab 175 MB failile ning keskmised logifailid on sellest mahukamad.

2) Suurte andmehulkade korral on andmetöötlus aeglane.

Maksimaalse mahutavuse juures (1 048 576 rida) on andmete filtreerimine ja graafikute väljajoonistumine aeglasel. Iga kord kui vahetatakse komponenti ja mõõdikut, mille kohta graafikut joonistada, arvutatakse graafiku loomise jaoks kõigi andmete põhjal uuesti väärtused ning seetõttu võtab see tavalise 2-3 sekundi asemel kuni 10 sekundit.

3) Imporditud andmete korduvkasutamine on keeruline.

Korduvkasutamise võimaluseks on salvestada andmed Exceli failina iga kord või eksportida filtreeritud andmed, mis iga kord uuesti sisse lugeda tuleks. Nii esimese kui ka teise variandi puhul tuleks hallata paljusid faile ja mõlema puhul esineb probleem, mis on kirjeldatud selle peatüki punktis 4.

4) Keeruline implementeerida pika ajavahemiku andmete kuvamist ühel graafikul.

30 faili impordil on raske filtreerida esimese Exceli tabeli (*sheet*) peale ainult vajalikke mõõdikuid. Põhjus on selles, et kui eraldada tabelitesse erinevad mõõdikud, siis kasvab tabelite arv meeletult kiiresti ja nende haldamine on problemaatiline. Kui hoida kõiki andmeid ühes tabelis, siis tekib probleem, mis on kirjeldatud punktis 1.

5) Kasutajasõbraliku ja mugava kasutajaliidese loomine on keeruline.

Excel pakub võimaluse kasutada vaid limiteeritud valikut kasutajaliidese elemente. Neid ei saa kasutaja ise disainida ega piisavalt paindlikult positsioneerida.

3. Analüüs

Praegune lahendus ei ole kliendi jaoks sobilik ning seetõttu paneb autor järgmises alampeatükis kirja soovitud lahendusele esitatavad nõuded. Nõuded on väljatöötatud koostöös kliendi meeskonnaga, kes peaksid hakkama seda lahendust oma töös kasutama.

3.1 Nõuded

Nõuete analüüsil lähtub autor FURPS+ analüüsi raamistikust [37]. Nõuete kirja panemisel on silmas peetud hea nõude kolme baasomadust: ühene kontrollitavus, kerge kontrollitavus, sõnastuse lihtsus ja lühidus [31].

Funktsionaalsus (*Functionality*)

- Rakendusse peab saama laadida logifaile, mille maht on kuni 300 MB või mis sisaldavad kuni 1 850 000 rida.
- Rakendus peab toetama nii Unixi/Linuxi („\n“) kui ka Windowsi („\r\n“) reavahetussümboleid logifailides.
- Rakendus peab suutma logi reast eraldada info: domeeni nimi, *MF Containeri* nimi, komponendi nimi, mõõdiku nimi, mõõdiku väärtus, ajatempel.
- Mõõdiku nimetuse alusel peab rakendus suutma tuvastada, mis tüüpi komponendiga on tegemist. Tüübiks on *MF Container*, kui nimetus algab sõnega: „system.“. *Broker*, kui nimetus algab sõnega: „broker.“. *ESB Container*, kui nimetus algab sõnega: „*esb.service.messages*.“.
- Rakendusse peab saama sisestada kliendi nime või valida kliendi nime juba sisestatud klientide loendist.
- Logifaili rida peab vastama järgnevale mustrile:

```
[Logisse kirjutamise kuupäev ja kellaaeg] [logi rea tüüp]
source=[Domeeni nimi].[MF Containeri nimi] :ID=[Komponendi
nimi] ;id=[mõõdiku nimi];value=[mõõdiku
väärtus];timestamp=[ajatempel]
```

- Filtreerida peab saama komponendi nime, mõõdiku nime ja ajavahemiku järgi.
- Rakendus peab töötama veebilehitsejates Firefox (versioon 35 ja uuemad) ja Chrome (versioon 40 ja uuemad).
- Rakendus peab säilitama ja välja näitama üles laetud failide nimed.
- Iga komponendi (*MF Container, Broker, ESB Container*) iga mõõdiku kohta peab saama luua joograafiku, kus on ainult selle mõõdiku kohta käivad andmed.

Kasutatavus (*Usability*)

- 75% kasutajatest peavad suutma ilma eelneva juhendamiseta laadida rakendusse faili ja luua graafiku neid huvitava komponendi mõõdiku kohta.
- Kõik kasutajad peavad suutma laadida 60 MB faili rakendusse ning valida neid huvitava mõõdiku ning ajaperioodi vähem kui 2 minutiga. Kasutajateks on kliendi juures töötavad eriala eksperdid.
- Kasutajaliides peab olema inglisekeelne.

Käideldavus (*Reliability*)

- Rakenduse tavapärase töö taastamisele kuluv aeg on maksimaalselt 2 töönädalat.

Jõudlus (*Performance*)

- 43 200 (ühe kuu mõõdikute kogus ühe minutise mõõtmise intervalli juures) andmepunkti peab joonistuma välja maksimaalselt 10 sekundiga.
- Rakendus peab suutma üles laadida ja andmebaasi sisestada 60 MB faili keskmiselt 10 sekundiga ja maksimaalselt 15 sekundiga.

Toetatavus (*Supportability*)

- Mitte ühegi rakenduse osa arendus ei tohi olla oma elutsükli lõppfaasis (*end-of-life*) [49].

+ (disain, tehnilise realiseerimise piirangud, liideste piirangud, majutuse piirangud, füüsilised piirangud jms)

- Server, kus rakendus asub, peab asuma rakenduse tellija sisevõrgus, kuhu pääseb ligi ainult tellija kohalikust võrgust või läbi VPN-i. Andmebaas peab asuma samuti tellija sisevõrgus.
- Välise kliendi andmeid tohib hoida ainult rakenduse tellija sisevõrgus asuvas andmebaasis. Väliseks kliendiks on rakenduse tellija klient, kelle andmeid analüüsitakse.
- Kõik väärtustatud andmeväljad peavad olema kasutajaliidesel *drop-down*iga realiseeritud.
- Graafikud peavad olema loetavad 17 tolliste ja suuremate monitoride pealt.

3.2 Võimalike lahenduse osade kirjeldus

Analüüsidest võimalikke lähenemisviise probleemi lahendamiseks, otsustas autor jagada võimaliku lahenduse kaheks alamosaks. Alamosad on teineteisest suuresti sõltumatud.

Osad, mida on vaja tulevases lahenduses:

- 1) Logifailide sisselugemine. Info grupeerimine komponentide ja mõõdikute põhiselt.
- 2) Saadud andmete põhjal graafiku joonistamine.

Autor leiab, et on olemas neli võimalikku kombinatsiooni, millest lähtuda uue lahenduse loomisel.

Võimalused on:

- 1) Kasutada mõlema osa jaoks mõnda olemasolevat rakendust.
- 2) Arendada ise mõlemad osad.
- 3) Arendada osa 1 ja kasutada osa 2 jaoks mõnda olemasolevat rakendust.
- 4) Arendada osa 2 ja kasutada osa 1 jaoks mõnda olemasolevat rakendust.

3.3 Logifailide töötlemiseks mõeldud rakenduste võrdlus

Tabel 1 Logifailide töötlemiseks mõeldud rakendused

Rakendus	Failide sisselugemine	Failide eksport (CSV, XSL)	Komponentide ja mõõdikute põhine eraldamine eraldi failidesse	Ajatempli muutmine inimloetavasse kuupäevaformaati	Korduv uuestiseadistamine	Suudab töödelda faile rohkem kui 1,85 miljoni reaga
LibreOffice [43]	+	+	+	+	+	-
Microsoft Excel [27]	+	+	+	+	+	-
DataParse [29]	+	+	+	-	+	+
LogQL [25]	+	+	-	-	+	+
ParseRat [8]	+	+	+	-	-	-

Roheline värv tabelis 1 tähendab kriteeriumile vastavust ja kollane tähendab kriteeriumile mittevastavust. Tabelis 1 võrreldud rakendustes hindas autor ainult nende logide töötlemisega seonduvaid võimalusi. Failide sisselugemise kriteeriumi all hindas autor rakenduse võimet lugeda sisse tekstifaile. Failide ekspordi kriteeriumi all hindas autor rakenduse võimet eksportida faile CSV ja XSL (Exceli) formaadis. Komponentide ja mõõdikute põhise eraldamise kriteeriumi all hindas autor rakenduse võimet eraldada sisseloetud andmetest komponendid ning omakorda iga komponendi kohta eraldada mõõdikud. Ajatempli inimloetavasse kuupäevaformaati muutmise kriteeriumi all hindas autor rakenduse võimet muuta ajatemplid kuupäevaks ja kellaajaks. See kriteerium on tähtis, kuna ilma kuupäeva ja kellaajata ei saa andmeid analüüsiv ekspert konkreetseid järeldusi teha. Korduva uuestiseadistamise kriteeriumi all hindas autor rakenduse võimet hoida meeles kasutaja poolt info eraldamiseks määratud sätteid. 1,85 miljoni rea töötlemise suutlikkuse hindamine on tähtis, kuna logi failid suurimates domeenides sisaldavad nii palju ridu.

Olemasolevatest rakendustest ei ole ükski täielikult sobiv. LibreOffice'i miinuseks on töödeldavate ridade piirang, mis on samamoodi ka Exceli miinuseks. Exceli probleemid on lisaks täpsemalt kirjeldatud peatükis 2.3. LogQL suutis küll väärtused eraldada failidest, kuid neid peaks sorteerima käsitsi, kasutades SQL-i laadset keelt ning puudu on ka ajatempli muutmise tugi. DataParse'i probleemiks on ajatempli muutmise toe puudumine. ParseRati peamisteks puudusteks on tegevuste korduva tegemise vajadus ja töödeldavate ridade arvu kriteeriumile mittevastavus. Sellest tulenevalt ei sobi ükski siinkohal uuritud ja võrreldud rakendus

probleemi

lahendama.

3.4 Graafikute loomiseks mõeldud rakenduste võrdlus

Tabel 2 Graafikute loomiseks mõeldud rakendused

Rakendus	Faili sisselugemine (CSV, XSL)	Faili eksport (JPEG, PDF, PNG, GIF)	Automaatne pealkirjade määramine	Kuvab 43 200 andmepunkti 10 sekundiga	Korduv uuestiseadistamine
Microsoft Excel [27]	+	+	+	+	+
SigmaPlot [33]	+	+	+	+	+
LibreOffice [43]	+	+	+	-	+
Cacti [42]	-	+	+	+	+
Meta-Chart [51]	-	+	+	-	+
amCharts [2]	+	+	-	+	-
Online Chart Tool [36]	+	+	-	-	-
Rich Chart Live [5]	+	-	-	+	-
ChartGo [6]	-	+	-	-	-

Roheline värv tabelis 2 tähendab kriteeriumile vastavust ja kollane tähendab kriteeriumile mittevastavust. Võrdlusest tabelis 2 on välja jäetud kõik *Javascripti* teegid. Põhjuseks on see, et autor otsis rakendust, mida saaks kasutada ilma, et seda peaks laiendama. Sooviks oli kasutada graafikute loomiseks rakendust terves funktsionaalsuses andes ette ainult graafiku loomiseks vajalikud andmed. Failide sisselugemise kriteeriumi all hindas autor rakenduse võimet lugeda sisse CSV ja XSL (Exceli) formaadis faile. Faili ekspordi kriteeriumi all hindas autor rakenduse võimet eksportida graafikuid erinevates pildi formaatides. Automaatse pealkirjade määramise kriteeriumi all hindas autor rakenduse võimet määrata sisseloetud andmete põhjal graafikute pealkirju nii telgedele kui ka graafikule endale. 43 200 mõõdikut 10 sekundiga kuvamise kriteeriumit on vaja, kuna ühe minutise mõõtmise intervalli juures tekib just nii palju mõõdikuid ühes kuus. Korduva uuestiseadistamise kriteeriumi all hindas autor rakenduse võimet hoida meeles kasutaja poolt graafikute loomiseks määratud sätteid. Kuigi Excel ja SigmaPlot on võrdluse põhjal sobivad, on peatükis 4.1 punktis 2.5 toodud probleemid, mis tekivad nende rakenduste kasutamisel. LibreOffice ei suuda ühe kuu andmeid kuvada graafikul vähem kui 10 sekundiga ning sellise andmekoguse juures esineb tõsiseid puudujääke kogu rakenduse töös. Cacti ei sobi kuna see rakendus on mõeldud peamiselt võrgu andmete pärimiseks ning seetõttu ei ole tal failide sisselugemise tuge. Kõigil teistel võrreldud rakendustel on kaks või rohkem puudust. Sellest tulenevalt ei sobi ükski uuritud ja võrreldud rakendus probleemi lahendamaks.

4. Lahendus

Järgnevalt kirjeldab autor, kuidas loodi nõuetele vastav rakendus. Autor kirjeldab metoodika, mida järgis, ning esitab loodud rakenduse kirjelduse. Autor kirjeldab suurimat esinenud probleemi: lahenduse nõuetele vastava kiiruse saavutamine. Lisaks esitab autor kliendi meeskonnalt saadud tagasiside loodud rakenduse kohta ning edasised rakenduse arendused ja plaanid. Autor on teinud rakenduse lähtekoodi kättesaadavaks kaitsmise protsessi ajaks lisa 1 all oleval aadressil. Vastavalt lähtekoodiga kaasa pandud litsentsiteatele võib lähtekoodi kasutada vaid hindamise protsessi jaoks. Programmeerimisel on autor lähtunud *clean code* koodi kirjutamise eeskirjadest.

4.1 Metoodika

Järgnevalt esitab autor punktis 3.2 toodud võimalike kombinatsioonide analüüsi punktides 3.3 ja 3.4 võrreldud rakenduste põhjal.

1) Kasutada mõlema osa jaoks mõnda olemasolevat rakendust.

Peatükist 3.3 selgub, et ükski logifailide töötlemiseks mõeldud rakendus nõutud kriteeriumitele ei vasta ning peatükis 3.4 toodud järeldustest selgub, et ükski graafikute loomise rakendus ka ei sobi. Seega ei saa mõlema osa jaoks kasutada mõnda olemasolevat rakendust.

2) Arendada osa 1 ja kasutada osa 2 jaoks mõnda olemasolevat rakendust.

Kui kirjutada ise logide töötlemise osa ning kasutada tabelis 2 võrreldud rakendusi, tekivad ikkagi järgnevalt toodud probleemid. Järgnevad väited, v.a viimane, on tehtud eeldusel, et hinnatakse ainult graafiku loomise funktsionaalsust, mitte ka andmete töötlemist nagu võimaldavad Excel [27] ja LibreOffice [43].

- 1) Ühe kuu logifailidest tekib 100 grupeeritud andmeid sisaldavat faili, keskmise suurusega kliendi domeeni puhul, kui need unikaalse komponendi ja mõõdiku põhjal eraldada.
- 2) Need 100 faili tuleb ükshaaval sisse laadida graafiku loomise rakendusse.

- 3) Tekib 100 graafikut, mida hoitakse failisüsteemis, kui mitte tahta iga kord uuesti andmeid sisse lugeda ja graafikut luua.
- 4) Puudub paindlikkus lisada varasema või hilisema perioodi andmeid juba loodud graafikutele. Selleks, et seda oleks võimalik teha, tuleb andmeid uuesti töödelda.
- 5) Selleks, et vaadata pikema perioodi kui 1 kuu mõõdikuid peab:
 - a. Logide töötlemise rakendus suutma rohkem kui 6 GB faile töödelda, kuna ühe kuu kohta genereeritakse keskmiselt sellises mahus logifaile.
 - b. Rakendus uuesti töötlemata juba kord töödeldud logifailides asuvaid andmeid. Selle efektiivseks lahendamiseks tuleb kuidagi säilitada juba kord töödeldud andmed, kas siis failis või andmebaasis. Seda arendust ei ole vaja, kui juba kord töödeldud info salvestatakse andmebaasi. Sellest tulenevalt, kui juba logide töötlemise rakendus kasutab andmebaasi ja selle põhjal graafiku rakendusele faile loob, siis on efektiivsem teha täiendav arendus ning kasutada graafikute loomiseks mõnda tasuta *Javascripti* teeki. *Javascripti* teeki kasutava koodi muutmine on lihtsam ja paindlikum kui olemasolevate rakenduste autoritele muutmise palvete esitamine. Lisaks sellele oleks kasutajaliidese loogikat realiseeriv kood ka muudetav. Sellest tulenevalt saab teha muudatusi ja uusi arendusi sõltumatult graafiku rakendust arendavast firmast.
- 6) Tülikas on vaadata sama komponendi kahte erinevat mõõdikut, kui selleks peab vastava faili kõigepealt üles leidma ja siis uue sisselugemise tegema.
- 7) Kasutades lisaks LibreOffice'i ja Exceli lisafunktsionaalsust, andmete töötlemist, hakkab nende rakenduste failide või tabelite arv kasvama samamoodi nagu eelpool toodud graafikute salvestamise näites ning logide töötlemise rakendus peab suutma lisaks tekstifailidele töödelda ka Exceli või LibreOffice'i faile. Lisaks ei suuda kumbki rakendus 200 MB logifaile töödelda ja seetõttu vajavad andmete eelnevat töötlemist.

3) Arendada osa 2 ja kasutada punkti 1 jaoks mõnda olemasolevat rakendust.

Peatüki 3.3 põhjal ei vasta ükski logifailide töötlemiseks mõeldud rakendus nõutud kriteeriumitele ja seetõttu ei ole see kombinatsioon võimalik.

4) Arendada ise mõlemad osad.

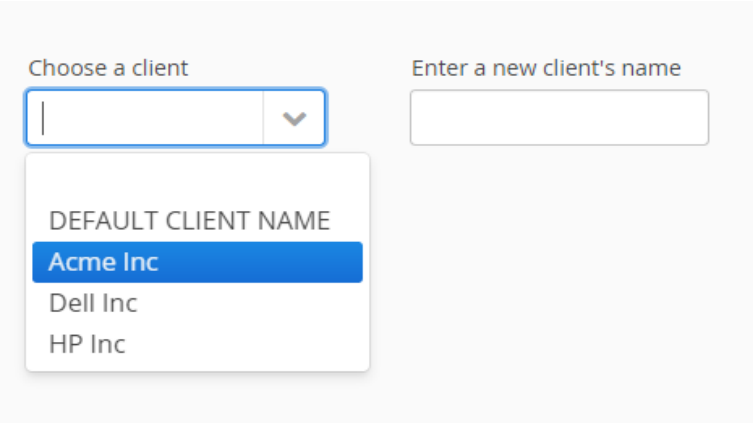
Kuna eelnevate osade kombinatsioonidega ei olnud võimalik koostada nõuetele vastavat lahendust peab autor arendama mõlemad osad.

4.2 Rakenduse kirjeldus

Rakenduse võib jaotada kaheks osaks: veebiraamistiku ja andmete hoiustamise osaks. Veebiraamistikuna kasutas autor Java veebiraamistikku Vaadin 7.4 [48]. Veebiraamistiku valikul lähtus autor kahest raportist [34], [35], mille on avaldanud Java tehnoloogiatega tegelev firma Zereturnaround. Andmete hoiustamiseks kasutab autor andmebaasisüsteemi PostgreSQL 9.3 [46]. Andmebaasisüsteemi valik on põhjendatud täpsemalt järgmises alampeatükis. Rakenduse sõltuvuste ja ehitamise haldamiseks kasutab autor tarkvara Maven [40] ning logimiseks vabavaralist teeki Apache log4j [39]. Rakendus on paigaldatud Java rakendusserverile Tomcat 7 [41].

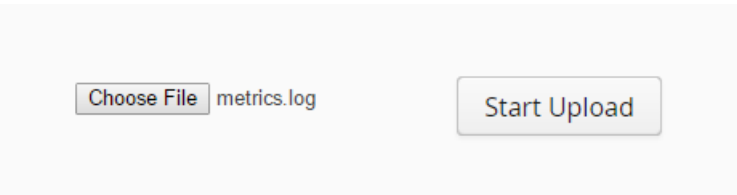
Funktsionaalsused, mis on rakenduses realiseeritud:

Olemasoleva kliendi valimine nimekirjast või uue kliendi sisestamine on esitatud joonisel 2. Olemasolevate klientide valik on implementeeritud *drop-downi* abil ning uut klienti saab sisestada kasutades tekstilahtrit.



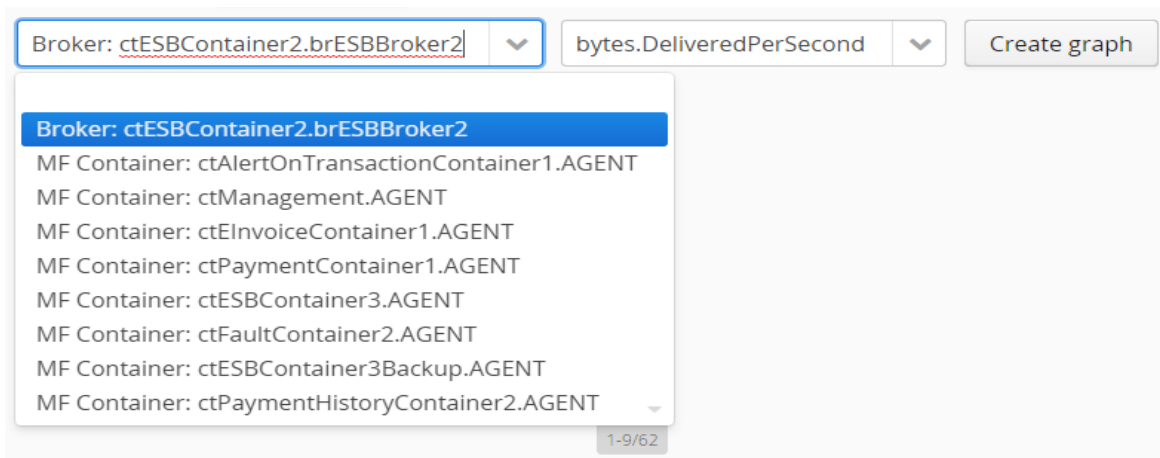
Joonis 2 Olemasoleva kliendi valimine nimekirjast või uue kliendi sisestamine

Logifailide üles laadimine ja andmebaasi sisestamine on esitatud joonisel 3.



Joonis 3 Logifailide üles laadimine ja andmebaasi sisestamine

Komponendi ja mõõdiku põhine filtreerimine on esitatud joonistel 4 ja 5. Vastavalt kasutajate esitatud nõudele on süsteemis eksisteerivad väärtused esitatud *drop-downis*.

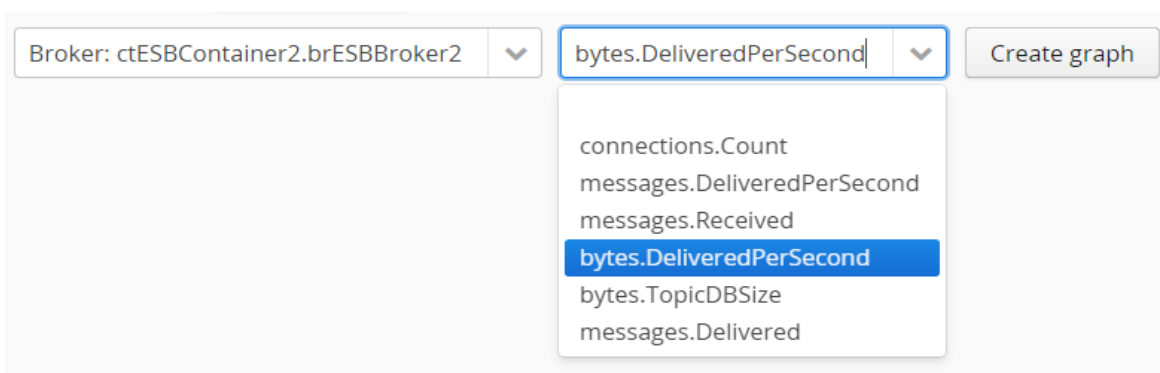


Broker: ctESBContainer2.brESBBroker2 | bytes.DeliveredPerSecond | Create graph

- Broker: ctESBContainer2.brESBBroker2
- MF Container: ctAlertOnTransactionContainer1.AGENT
- MF Container: ctManagement.AGENT
- MF Container: ctEInvoiceContainer1.AGENT
- MF Container: ctPaymentContainer1.AGENT
- MF Container: ctESBContainer3.AGENT
- MF Container: ctFaultContainer2.AGENT
- MF Container: ctESBContainer3Backup.AGENT
- MF Container: ctPaymentHistoryContainer2.AGENT

1-9/62

Joonis 4 Komponenti põhine filtreerimine



Broker: ctESBContainer2.brESBBroker2 | bytes.DeliveredPerSecond | Create graph

- connections.Count
- messages.DeliveredPerSecond
- messages.Received
- bytes.DeliveredPerSecond
- bytes.TopicDBSize
- messages.Delivered

Joonis 5 Mõõdiku põhine filtreerimine

Ajaperioodi põhjal filtreerimine on esitatud joonisel 6. Ajaperioodi valimise mugavdamiseks on kasutatud kalendri pistikprogrammi (*plugin*).



Please select the Time period (default is max):

June 2014

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

05 : 02 : 03 PM

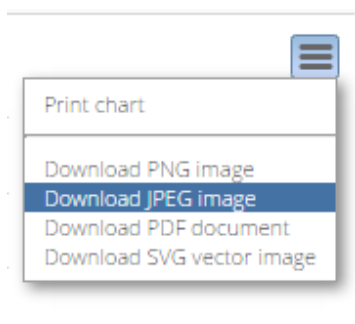
Joonis 6 Ajaperioodi põhjal filtreerimine

Üles laetud failide nimekirja nägemine on esitatud joonisel 7.

```
Files uploaded for this client:  
metrics.log.2014-06-15  
metrics.log.2014-06-16  
metrics.log.2014-06-17  
metrics.log.2014-06-19  
metrics.log.2014-06-20
```

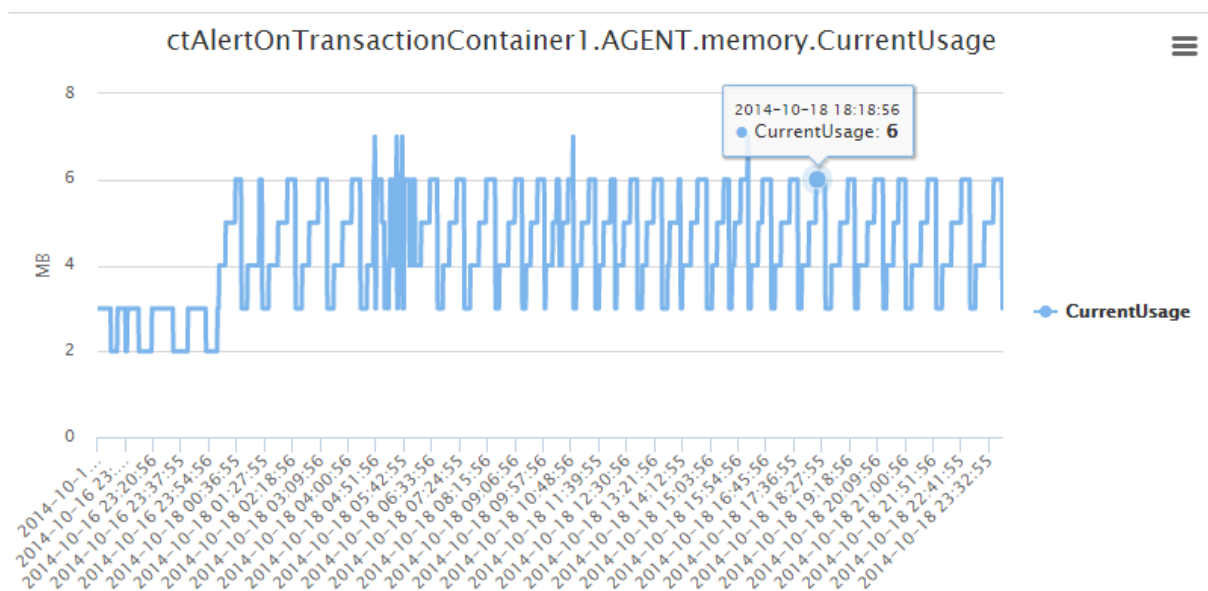
Joonis 7 Üles laetud failide nimekiri

Graafiku printimine ja graafiku eksport on esitatud joonisel 8.



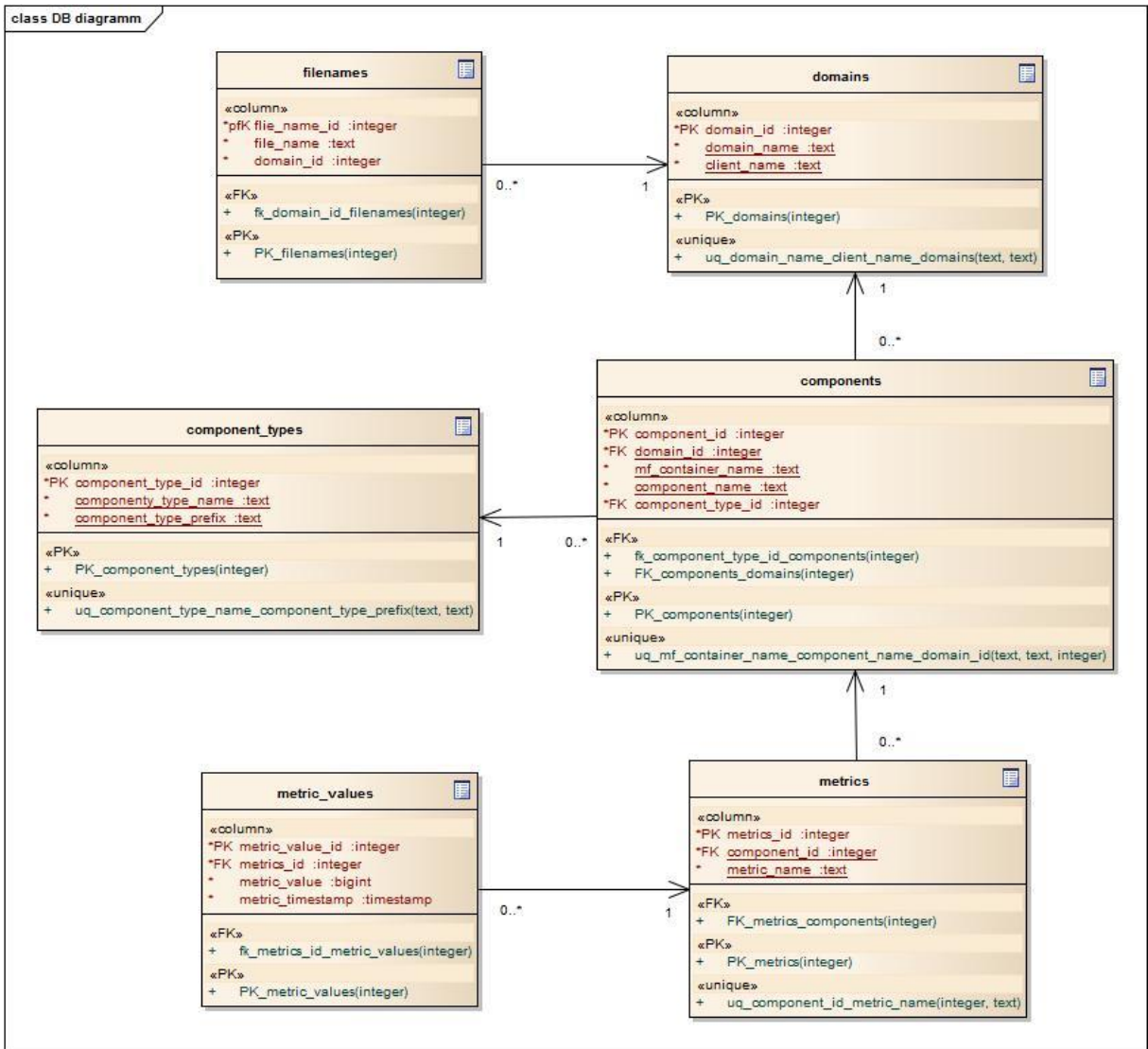
Joonis 8 Graafiku printimine ja graafiku eksport

Graafiku kuvamine on esitatud joonisel 9. Graafiku loomiseks on kasutatud Highcharts [24] Javascripti teeki.



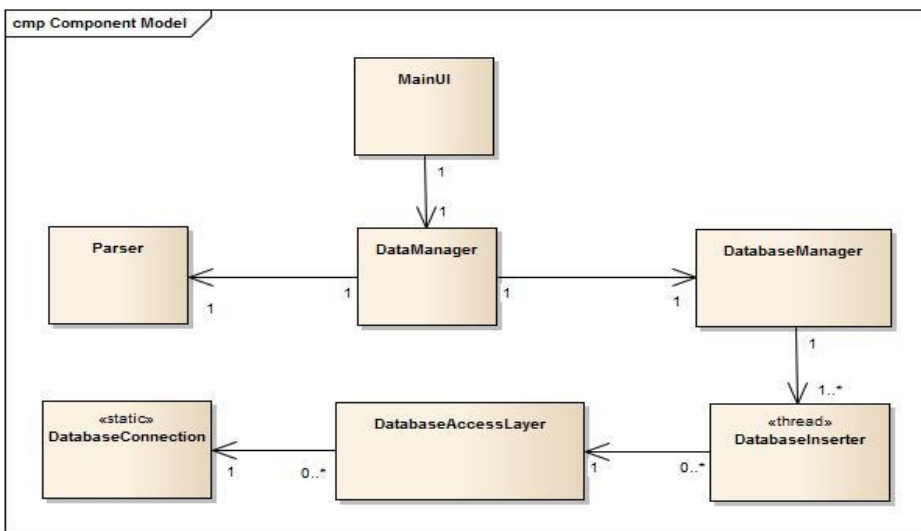
Joonis 9 Graafiku kuvamine

Joonisel 10 on toodud kasutatav andmebaasiskeem.



Joonis 10 Andmebaasiskeem

Joonisel 11 on toodud rakenduse ülesehituse komponentmudel.



Joonis 11 Rakenduse komponentmudel

Rakenduses on kokku 2082 rida seitsmes klassis, millest 388 on tühjad read, 13 on kommentaaride read ja 1681 on koodiread.

MainUI – MainUI on kasutajaliidese loomise ja kasutaja tegevustele reageerimise loogikat realiseeriv klass. Lisaks tekstiväljade, *drop-downide* ja tekstide loomisele on siin realiseeritud ka failide üles laadimine. Failide üles laadimine on siin klassis, kuna pool implementeeritud koodist on seotud kasutajaliidese elementidega ning teine pool vajab ligipääsu siin klassis paiknevatele muutujatele. Selleks, et vältida tarbetut objektide loomist ja objektide edastamist, otsustas autor hoida seda funktsionaalsust siin.

DatabaseInserter – DatabaseInserter on lõime klass. Sellele klassile antakse ette ridade kogus, mida ta sisestama peab hakkama. Enne töötlust teeb lõim kindlaks, mis tarkvara versioonist need read pärit on ja määrab vastavalt selle regulaaravaldise, mida ridade töötlemisel lõim kasutab. Samuti määratakse siin, mis tüüpi sisestatav komponent on ning eemaldatakse komponendi tüübi järgi vastava moodsiku nimest üleliigne prefiks („*system.*“, „*broker.*“ või „*esb.service.messages*“).

DataManager – DataManager klass sisaldab endas logi ridade töötlemise ja andmebaasi sisestamise haldamist. DataManager delegeerib vastavale alamklassile tegevuse õigel ajal. Enne, kui logi ridade töötlemisega tegelevale klassile kontroll antakse, leitakse siin klassis ridade arv ning sellest tulenevalt jaotab ridade töötlemisega tegelev klass read ära. Kui read on loenditesse jaotatud, on järg andmebaasi sisestava klassi käes. See klass annab igale DatabaseInserter klassile ette ühe loendi, mida see hakkab andmebaasi sisestama. DataManager realiseerib ka lõimede lõpetamise kontrolli. Kuna lõimed lõpetavad erinevatel aegadel, siis on tähtis, et minnakse edasi alles siis, kui kõik lõimed on lõpetanud.

Parser – Parser klass sisaldab endas kogu loogikat logi ridade sisselugemise kohta. Parser klassile antakse ette mitmesse loendisse peab ta kogu info jagama ning mitu rida igasse loendisse läheb. Loend koosneb sõnade massiividest, mis sisaldavad ettemääratud koguse ridu.

DatabaseManager – DatabaseManager klass tegeleb andmebaasi andmete sisestamise ja kontrolliga. DatabaseManager tegeleb eraldi domeeni, komponentide ning moodsikute

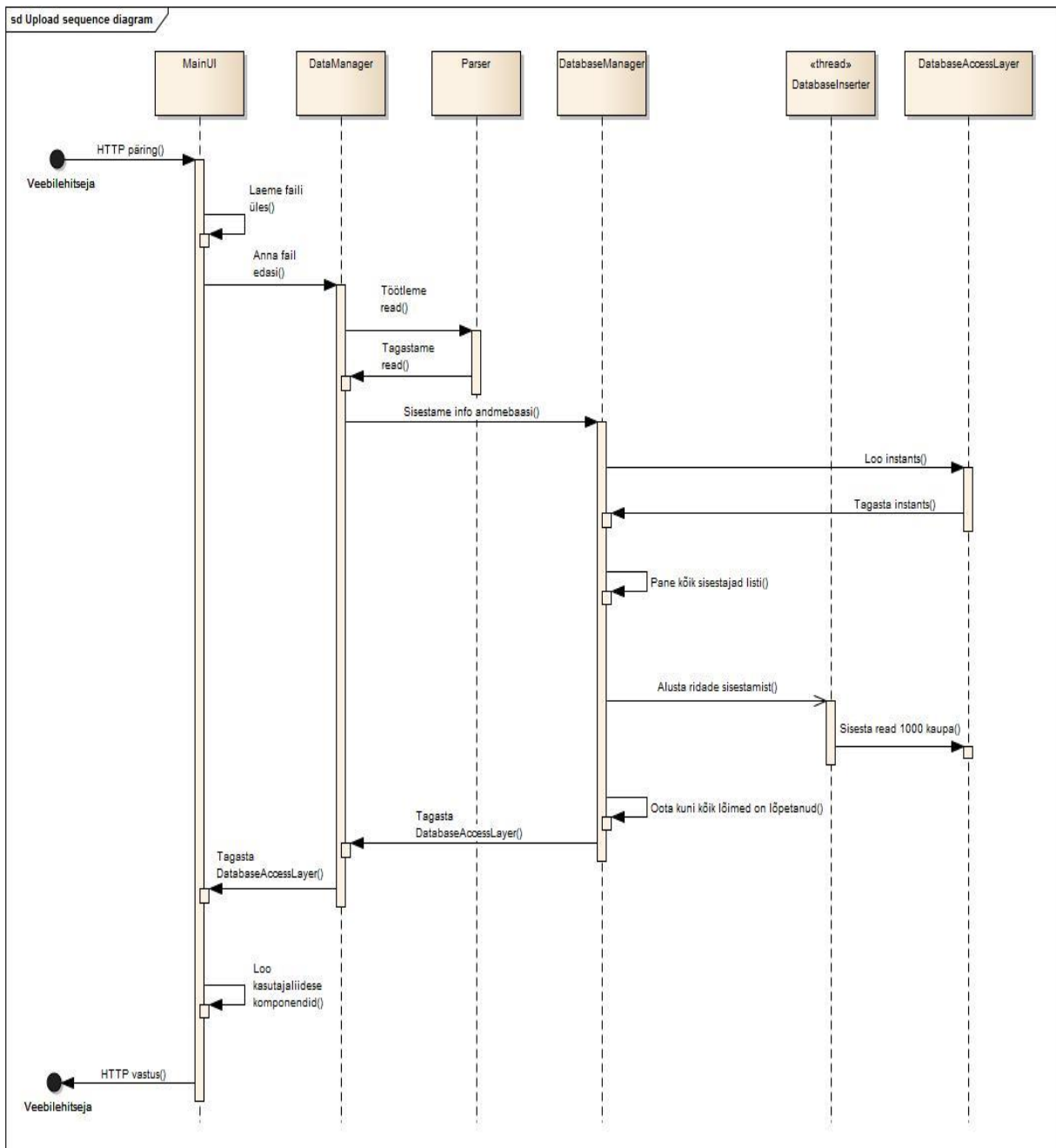
sisestamisega. Igaühe jaoks kasutatakse DataAccessLayer klassis olevaid andmebaasiga ühendumiseks mõeldud meetodeid.

DatabaseAccessLayer – DatabaseAccessLayer klass tegeleb suhtlusega, mis toimub andmebaasi ja rakenduse vahel. Kõik päringud, mis andmebaasi tehakse, käivad läbi selle klassi. Nii andmete sisestamised lõimed poolt, kui ka andmete pärimine graafiku jaoks.

DatabaseConnection – DatabaseConnection klass sisaldab andmebaasiga ühendumiseks vajalikku infot. DatabaseAccessLayer pöördub selle klassi poole, et saada andmebaasi ühendus.

4.3 Kiiruse saavutamine

Käesolevas peatükis kirjeldab autor, kuidas saavutati rakenduse töö nõutud kiirus. Joonisel 12 on esitatud faili üleslaadimise ja andmebaasi sisestamise protsessi jadadiagramm. See on peamine protsess, mida autor optimeeris.



Joonis 12 Faili üleslaadimise ja andmebaasi sisestamise protsessi jadadiagramm

Faili üleslaadimise ja andmebaasi sisestamise protsess koosneb peamiselt kahest osast: faili üles laadimine ja mällu lugemine ning andmete andmebaasi sisestamine. Kogu protsess osutus aeglaseks ja seetõttu hakkas autor otsima viise, kuidas seda kiirendada. Esimeseks lüliks selles protsessis on faili üles laadimine ja mällu lugemine. Kuna see osa võtab koguprotsessist ainult 1,1 – 2,5 % ajast (vaata Tabel 3) ja autor kasutas Java standard teegi meetodit kõigi ridade lugemiseks, siis autor seda osa protsessist optimeerima ei hakanud.

Tabel 3 Faili mällu lugemise ja andmete andmebaasi sisestamise osakaalu võrdlus

Ridade arv	Faili mällu lugemine ja ridade loendisse panemine (sekundit)	Osakaal kogu protsessi ajast (protsendina)	Andmete sisestamine andmebaasi kümne lõimega (sekundit)	Osakaal kogu protsessi ajast (protsendina)
100 000	0,12	2,5	4,68	97,5
350 000	0,17	1,68	9,93	98,32
700 000	0,38	1,55	24,05	98,45
1 150 000	0,54	1,12	47,50	98,88

Teine osa protsessist on andmebaasi andmete sisestamine. Andmete sisestamise kiiruse tõstmiseks proovis autor kasutada mäluandmebaasi. Autor otsis enim kasutatavaid vabavaralisi mäluandmebaase ning leidis andmebaasid HSQLDB [44] ja H2 [45]. Nende andmebaaside probleemiks sai aga see, et nad ei toeta pakktöötlust (*batch insert*) [9] ning seetõttu oli andmete andmebaasi sisestamine ikkagi aeglane. Lisaks oli probleem selles, et pärast andmebaasi sulgemist kadusid ka andmed, nii et igapäevaseks kasutamiseks oleks vaja olnud ka andmete säilitamist. Kuna andmekogused, mida kasutatakse, on suured, siis otsustas autor otsida persistentset lahendust, mis ei vajaks iga kord andmete uuesti sisselugemist. Lõplikuks andmebaasiks valis autor PostgreSQL [46]. PostgreSQL kasuks otsustas autor, sest see andmebaasisüsteem ei ole sõltuv ühestki raamistikust, see on relatsiooniline andmebaas, see on vabavara, autor on varem kasutanud seda andmebaasisüsteemi ning kiiruse võrdluses [52] olid vabavaralised PostgreSQL ja MySQL [30] võrdsed. PostgreSQL kasutades saavutas autor suurema andmete sisestamise kiirus kasutades pakktöötlust (*batch insert*) ning luues välisvõtmete veergudele indeksid. PostgreSQL loob primaarvõtmete veergudele indeksid automaatselt. Eemaldades andmete sisestamise ajaks tabelist *metric_values* unikaalsuse indeksi, saavutas autor keskmiselt 11,8 sekundit ajalists võitu iga 200 MB faili kasutamisel. Autor tõstis ka PostgreSQL jagatud puhvri suurust 512 MB-ni, et andmebaasisüsteemil oleks suurem hulk mälu, mida kasutada puhverdamiseks. Samuti muutis autor PostgreSQL konfiguratsioonis parameetrit *effective_cache_size*, mida

andmebaasisüsteem kasutab hindamaks, millist täitmisplaani kasutada. Parameetri suuruseks valis autor 384 MB. Autor muutis ka parameetrit `checkpoint_segments`, mis määrab mitme transaktsiooni hoidmiseks mõeldud segmendi järgi PostgreSQL kontrollpunkti (*checkpoint*) loob. Väärtuseks valis autor 32 segmenti. Samuti lülitas autor välja parameetri `synchronous_commit`, mis võimaldas andmebaasisüsteemil teha sekundis rohkem püsivaid salvestusi (*commit*). Autor valis konfiguratsiooni muudatuste suurused vastavalt ametlikule PostgreSQL'i dokumentatsioonile. [47] Esitatud konfiguratsiooni muudatusega saavutas autor kasutades 200 MB faile keskmiselt 5,1 sekundit kiirema tulemuse. Keskmised tulemused on arvutatud kümne katse põhjal.

Selleks, et saavutada veel kiiremat tulemust, muutis autor rakenduse loogika lõimepõhiseks. Ridadest väärtuste eraldamine on viidud lõimede tasemele, et seda saaksid mitu lõime paralleelselt teha. Kui seda teha faili mällu lugemisel Parser klassis, siis seda teeks ainult üks lõim. Lisaks, et vältida üleliigset andmebaasi koormamist identifikaatorite päringute jaoks, on DatabaseManager klassis olemas lõimekindlad andmekonteinerid (*Concurrent Collection*), mida kõik lõimed jagavad. Enne, kui lõim ükstapuha millist väärtust hakkab andmebaasi sisestama, kontrollib ta kõigepealt ega talle vajalikke identifikaatoreid ei ole juba andmekonteineris olemas. Kui lõim ei leia talle vajalikku väärtust, alles siis sisestab ta andmebaasi uue kirje ning seejärel uuendab andmekonteineri sisu saadud identifikaatoriga. Selle tõttu jäi ära faili ridade arvu jagu (keskmiselt 1,15 miljonit) päringuid andmebaasi pihta. Nii domeenide, komponentide kui ka mõõdikute jaoks on eraldi andmekonteinerid. Need lõimekindlad andmekonteinerid on toodud DatabaseManager klassi, et kõik lõimed saaksid neid jagada.

Autor on kasutanud andmebaasiga suhtlemiseks ettevalmistatud lauseid (*prepared statement*) tekstikujul SQL-iga (*raw SQL*). Autor ei valinud mõnda ORM-i [38] andmebaasiga suhtlemiseks, sest see oleks tekitanud üleliigset objektide loomist. [1], [4], [26]

Tabel 4 Faili mällu lugemise ja andmete andmebaasi sisestamise ajaline võrdlus

Ridade arv	Faili mällu lugemine ja ridade loendisse panemine (sekundites)	Andmete sisestamine andmebaasi ühe lõimega (sekundites)	Andmete sisestamine andmebaasi kümne lõimega (sekundites)
100 000	0,12	11,56	4,68
350 000	0,17	30,78	9,93
700 000	0,38	82,18	24,05
1 150 000	0,54	164,8	47,50

Tabelist 4 järeldub, et kümne lõimega andmebaasi sisestamine on 2,5 – 3,5 korda kiirem kui ühe lõimega sisestamine.

4.4 Tagasiside rakenduse kasutajatelt

Tagasisidet küsiti kliendi neljalt eksperdilt saates järgnevalt esitatud küsimused emailile ning kasutajatel oli enne seda võimalus vabalt kasutada rakendust. Tagasiside andmine ei olnud anonüümne.

Kasutajatele esitatud küsimused:

- 1) Kas on midagi kasutajaliidese juures, mida sooviksite muuta?
- 2) Kas on midagi graafiku juures, mida sooviksite muuta?
- 3) Kui lihtne oli rakendust kasutada? (Laadida üles fail ja kuvada soovitud andmed graafikul)
- 4) Kas rakendus oli piisavalt kiire oma töös kasutamiseks?
- 5) Kas rakenduses oli midagi puudu, mida kindlasti sooviksite selles näha?

Tagasiside kasutajatelt:

- 1) Kasutajad avaldasid soovi, et kliendi loomise võiks rakenduse töös muuta eraldatumaks. Ettepanekuks tehti, et võiks nupule vajutamisega kinnitada kliendi loomise. Lisaks soovisid kasutajad, et oleks võimalik failide nimekirja kuidagi peita ja soovi korral nupule vajutusega näha. Üldine kasutajaliidese värvivalik ja väljanägemine oli kasutajate sõnul sobiv ja meeldiv.
- 2) Andmete kuvamine graafikule oli kõigi kasutajate jaoks ootuspärane ja sobiv.
- 3) Kõik kasutajad hindasid rakenduse kasutamist väga lihtsaks. Seda oli ka oodata, kuna tegemist on ekspertidega ning kontekstil põhinevate seoste loomine oli nende jaoks lihtne.
- 4) Kõik kasutajad hindasid, et rakenduse toimimine on piisavalt kiire, et seda oma töös kasutada.
- 5) Kasutajad sooviksid rakenduses näha võimalust lisada olemasoleva graafiku alla lisa graafikuid, võimalust kuvada samal graafikul analoogseid mõõdikuid erinevatelt komponentidelt, lisa filtreerimise võimalust *ESB Containeri* teenuste põhjal ning pirukas graafiku (*pie graph*) tugi *Broker* tüüpi komponendi mõõdikute osakaalu võrdlemiseks teiste sama tüüpi komponentidega.

4.5 Edasised arendused ja plaanid rakendusega

Autoril on plaanis rakendust aktiivselt edasi arendada. Lisaks eelmises peatükis toodud kasutajate soovide realiseerimisele kavatseb autor teha järgmisi tegevusi.

- 1) Lihtsustada ja restruktureerida olemasolevat koodi. Katta kogu rakendus testidega ning liikuda edasi kasutades testidest juhinduvat arendamist.
- 2) Lisada mitme faili korraga üles laadimise tugi.
- 3) Otsida kliendi firmas teisi tiime, kellele see rakendus võiks kasulik olla ning arendada ka nende soovitud funktsionaalsusi.
- 4) Luua andmete keskmistamise loogika, mille eesmärgiks on vähendada andmebaasi suurust ning teha suurte andmekoguste juures andmebaasisuhtlus kiiremaks.

- 5) Uurida veel täpsemalt andmebaasi konfigureerimise erinevaid võimalusi, et saavutada kiirem päringute tegemine.
- 6) Uurida võimalust reaalajas andmeid ESB-st pärida ning ka reaalajas neid kuvada.
- 7) Uurida võimalust muuta PostgreSQL mäluandmebaasiks, mis perioodiliselt salvestab andmeid kõvakettale.

5. Kokkuvõte

Töö põhieesmärgiks on leida lahendus logifailides asuvate andmete visualiseerimisele. Andmete visualiseerimist on vaja, et eksperdid saaksid loodud graafikute põhjal teostada analüüsi enda klientidele. Eesmärgi saavutamiseks on antud töös kirjeldatud järgmised sammud: praeguse lahenduse probleemide kaardistamine, ülevaade ja võrdlus eksisteerivatest rakendustest, analüüs ja nõuete väljatöötamine ning lahenduse loomine.

Kõik kasutajatega koos väljatöötatud nõuded said loodud rakenduses täidetud. Olulisemad nõuded, mis täideti, on 1,85 miljoni reaga failide impordi tugi, kasutaja poolt peaprotsessi sooritamine 60 MB failiga vähem kui kahe minutiga, ühe kuu andmete välja joonistumine graafikul vähem kui 10 sekundiga ja rakenduse poolt 60 MB faili üles laadimine ja andmebaasi sisestamine keskmiselt 10 sekundiga.

Suurte andmekoguste käsitlemise puhul tuleb palju rõhku panna protsessi efektiivsele täitmisele ning pudelikaelade ükshaaval eemaldamisele. Tähtis on keskenduda kõige rohkem aega võtvatele osadele. Aeglase protsessi osa väikene protsentuaalne paranemine annab märgatava muutuse lõpptulemuses. Näiteks nelja andmebaasi konfiguratsiooniparameetri muutmine andis 30% võitu kiiruses. Lisaks tõdeb autor, et tarkvaraarenduses on pideval analüüsil täpselt sama suur osakaal kui programmeerimisel.

Vastavalt kasutajatelt saadud tagasisidele on autoril plaanis arendada tugi mitme graafiku teineteise alla lisamiseks. Lisaks sellele soovib autor otsida kliendi firmas teisi tiime, kellele see rakendus võiks kasulik olla ning arendada ka nende soovitud funktsionaalsusi ja uurida võimalust reaalajas andmeid ESB-st pärida ning ka reaalajas neid kuvada.

Tähtsaimate lahendatud probleemide hulka kuulusid: töödeldavate andmeridade arvu piirang, andmetöötluse kiirus, töödeldud andmete korduvkasutatavus, ajavahemiku muutmise paindlikkus filtreerimiseks. Kliendi poolt püstitatud probleemid said lahendatud ning kõik esitatud nõuded on loodud rakenduse poolt rahuldatud. Loodud rakendusega on juba tehtud ka esimene töö kliendile ning tulemusega jäädgi rahule. Sellele tuginedes võib öelda, et püstitatud eesmärk saavutati.

Summary

The main objective of the thesis is to find a solution to the problem of visualizing data from log files. Data visualization is necessary for the client's team members to carry out analysis based on the graphs for their clients. In order to reach the goal, this thesis has described the following steps: analysing the problems of the current solution, overview of existing solutions and their comparison, analysis and requirements elicitation, creating the application.

All the requirements created together with the end users were satisfied by the created application. The most important requirements that were satisfied were: importing files with up to 1,85 million rows, completing the main process by the user with a 60 MB file in under two minutes, creating the graph with one month's data in under 10 seconds and completing the main process by the application with a 60 MB file on average in 10 seconds.

When processing large quantities of data, it is important to efficiently complete the process and remove the bottle-necks that slow the process down. It is important to focus on the parts of the process that take the most time. Even marginal improvement of a slow part of the process can give a significant return. For instance, changing four configuration parameters in the database gave a 30% improvement in speed. In addition, the author emphasises that continuous analysis throughout the development of an application is as important as the programming of the application.

Based on the feedback gained from the users of the application the author is planning on developing support to add other graphs under each other. In addition to that, the author intends to research the needs of other teams in the company and develop their requested functionalities as well. The author also intends to research the possibility of querying data directly from the ESB and creating the graph in real time.

Problems dealt with in the thesis include the following: the limit of rows that the application can process, the speed of operation, reuse of processed data and the flexibility of changing the time period for filtering. All the problems brought up by the client were solved and all the requirements were satisfied by the created application. A first task has already been completed using the created application. Based on the positive feedback the author claims that the initial goals were met.

Kasutatud kirjandus

- [1] A. Kornatsky, „Thoughts on SQL vs ORM,“ [Võrgumaterjal]. Available: <http://mindref.blogspot.com/2013/02/sql-vs-orm.html>. [Kasutatud 8 mai 2015].
- [2] amcharts.com, „AmCharts,“ amcharts.com, [Võrgumaterjal]. Available: <http://www.amcharts.com/>. [Kasutatud 26 veebruar 2015].
- [3] Aurea Software, „Create far better experiences with a far better Service Bus,“ Aurea Software, [Võrgumaterjal]. Available: <http://www.aurea.com/products/enterprise-service-bus>. [Kasutatud 5 mai 2015].
- [4] B. Karwin, „Why Should You Use an ORM?,“ [Võrgumaterjal]. Available: <http://karwin.blogspot.com/2009/01/why-should-you-use-orm.html>. [Kasutatud 8 mai 2015].
- [5] Blue Pacific Software, „Rich Chart Live,“ Blue Pacific Software, [Võrgumaterjal]. Available: <http://www.richchartlive.com/RichChartLive/default.aspx>. [Kasutatud 26 veebruar 2015].
- [6] ChartGo, „ChartGo,“ ChartGo, [Võrgumaterjal]. Available: <http://www.chartgo.com/>. [Kasutatud 26 veebruar 2015].
- [7] E. Heinsoo, „Avalik sektor pakub näidet läbimõeldud IT-süsteemist,“ Äripäev, 2006. [Võrgumaterjal]. Available: <http://www.aripaev.ee/uudised/2006/12/13/avalik-sektor-pakub-naidet-labimoeldud-it-susteemist>. [Kasutatud 4 aprill 2015].
- [8] Guy Software, „ParseRat,“ Guy Software, [Võrgumaterjal]. Available: <http://www.guysoftware.com/parserat.htm>. [Kasutatud 24 veebruar 2015].
- [9] H. Vallaste, „Batch processing,“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=524>. [Kasutatud 4 mai 2015].
- [10] H. Vallaste, „CSV (Comma Separated Values),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=2627>. [Kasutatud 12 aprill 2015].
- [11] H. Vallaste, „drop-down menu,“ [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=6107>. [Kasutatud 22 mai 2015].
- [12] H. Vallaste, „FTP (File Transfer Protocol),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=152>. [Kasutatud 12 aprill 2015].
- [13] H. Vallaste, „GB (Gigabyte),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=1166>. [Kasutatud 4 mai 2015].
- [14] H. Vallaste, „GIF (Graphics Interchange Format),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=155>. [Kasutatud 12 aprill 2015].
- [15] H. Vallaste, „JPEG (Joint Photographic Experts Group),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=199>. [Kasutatud 12 aprill 2015].
- [16] H. Vallaste, „JVM (Java Virtual Machine),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=3006>. [Kasutatud 12 aprill 2015].
- [17] H. Vallaste, „MB (Megabyte),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=1141>. [Kasutatud 4 mai 2015].

- [18] H. Vallaste, „PDF (Portable Document Format),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=1174>. [Kasutatud 12 aprill 2015].
- [19] H. Vallaste, „Plug-in, plugin,“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=266>. [Kasutatud 7 mai 2015].
- [20] H. Vallaste, „PNG (Portable Network Graphics),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=2342>. [Kasutatud 12 aprill 2015].
- [21] H. Vallaste, „RAM (Random-Access Memory),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=552>. [Kasutatud 12 aprill 2015].
- [22] H. Vallaste, „SSD (Solid State Drive),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=3395>. [Kasutatud 12 aprill 2015].
- [23] H. Vallaste, „VPN (Virtual Private Network),“ www.vallaste.ee, [Võrgumaterjal]. Available: <http://vallaste.ee/index.htm?Type=UserId&otsing=329>. [Kasutatud 12 aprill 2015].
- [24] Highcharts, „Highcharts,“ [Võrgumaterjal]. Available: <http://www.highcharts.com/>. [Kasutatud 13 mai 2015].
- [25] logQL, „logQL,“ logQL, [Võrgumaterjal]. Available: <http://www.logql.com/>. [Kasutatud 23 veebruar 2015].
- [26] M. Pronschinske, „The ORM Debate: Experts Weigh In,“ dzone.com, [Võrgumaterjal]. Available: <http://java.dzone.com/news/orm-debate-obsolete-technology>. [Kasutatud 8 mai 2015].
- [27] Microsoft, „Microsoft Excel,“ Microsoft, [Võrgumaterjal]. Available: <https://products.office.com/en-us/excel>. [Kasutatud 23 veebruar 2015].
- [28] MuleSoft Inc, „What is an ESB?,“ MuleSoft Inc, [Võrgumaterjal]. Available: <https://www.mulesoft.org/what-esb>. [Kasutatud 15 mai 2015].
- [29] National Data Parsing Canada Corporation, „Data Parse Business,“ National Data Parsing Canada Corporation, [Võrgumaterjal]. Available: <https://www.dataparse.com/business.aspx>. [Kasutatud 23 veebruar 2015].
- [30] Oracle Corporation, „MySQL,“ [Võrgumaterjal]. Available: <https://www.mysql.com/>. [Kasutatud 5 mai 2015].
- [31] P. Potter, „Süsteemi nõuete esiletoomine ja analüüs,“ Plumb, 2014. [Võrgumaterjal]. Available: <http://maurus.ttu.ee/sts/wp-content/uploads/2014/10/S%C3%BCsteemi-n%C3%B5uete-esiletoomine-ja-anal%C3%BC%C3%BCs.pdf>. [Kasutatud 2 märts 2015].
- [32] Spark59, Inc., „Minimum Viable Product,“ [Võrgumaterjal]. Available: <http://leanstack.com/minimum-viable-product/>. [Kasutatud 23 aprill 2015].
- [33] Systat Software Inc, „SigmaPlot,“ Systat Software Inc, [Võrgumaterjal]. Available: <http://www.sigmaplot.com/>. [Kasutatud 25 veebruar 2015].
- [34] Zereturnaround, „The 2014 Decision Maker’s Guide to Java Web Frameworks,“ [Võrgumaterjal]. Available: http://pages.zereturnaround.com/RebelLabs-AllReportLanders_The2014DecisionMakersGuidetoJavaWebFrameworks.html. [Kasutatud 7 veebruar 2015].
- [35] Zereturnaround, „The Curious Coder's Guide to Java Web Frameworks,“ [Võrgumaterjal]. Available: http://pages.zereturnaround.com/RebelLabs-AllReportLanders_TheCuriousCodersGuidetoJavaWebFrameworks.html. [Kasutatud 7 veebruar 2015].

- [36] Zygomatic, „onlineChartTool,“ Zygomatic, [Võrgumaterjal]. Available: <http://www.onlinecharttool.com/>. [Kasutatud 27 veebruar 2015].
- [37] T. L. J. Ottinger, „Agile in a Flash,“ 2009. [Võrgumaterjal]. Available: <http://agileinaflash.blogspot.com/2009/04/furps.html>. [Kasutatud 4 märts 2015].
- [38] TechTarget, „object-relational mapping (ORM),“ [Võrgumaterjal]. Available: <http://searchwindevelopment.techtarget.com/definition/object-relational-mapping>. [Kasutatud 23 aprill 2015].
- [39] The Apache Software Foundation, „Apache log4j 1.2,“ [Võrgumaterjal]. Available: <http://logging.apache.org/log4j/1.2/>. [Kasutatud 6 mai 2015].
- [40] The Apache Software Foundation, „Maven,“ The Apache Software Foundation, [Võrgumaterjal]. Available: <http://maven.apache.org/>. [Kasutatud 21 jaanuar 2015].
- [41] The Apache Software Foundation, „Tomcat 7,“ The Apache Software Foundation, [Võrgumaterjal]. Available: <https://tomcat.apache.org/download-70.cgi>. [Kasutatud 8 mai 2015].
- [42] The Cacti Group, „Cacti,“ The Cacti Group, [Võrgumaterjal]. Available: <http://www.cacti.net>. [Kasutatud 25 veebruar 2015].
- [43] The Document Foundation, „LibreOffice,“ The Document Foundation, [Võrgumaterjal]. Available: <https://www.libreoffice.org/>. [Kasutatud 23 veebruar 2015].
- [44] The hsql Development Group, „HSQLDB,“ [Võrgumaterjal]. Available: <http://hsqldb.org/>. [Kasutatud 5 mai 2015].
- [45] The HSQLDB Group, „H2 Database Engine,“ [Võrgumaterjal]. Available: <http://www.h2database.com/html/main.html>. [Kasutatud 5 mai 2015].
- [46] The PostgreSQL Global Development Group, „PostgreSQL,“ The PostgreSQL Global Development Group, [Võrgumaterjal]. Available: <http://www.postgresql.org/>. [Kasutatud 23 jaanuar 2015].
- [47] The PostgreSQL Global Development Group, „Tuning Your PostgreSQL Server,“ [Võrgumaterjal]. Available: https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server. [Kasutatud 22 mai 2015].
- [48] Vaadin Ltd, „Vaadin,“ Vaadin Ltd, [Võrgumaterjal]. Available: <https://vaadin.com/home>. [Kasutatud 12 jaanuar 2015].
- [49] Wikipedia, „End-of-life (product),“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/End-of-life_%28product%29. [Kasutatud 12 aprill 2015].
- [50] Wikipedia, „Enterprise service bus,“ [Võrgumaterjal]. Available: http://en.wikipedia.org/wiki/Enterprise_service_bus. [Kasutatud 13 mai 2015].
- [51] www.meta-chart.com, „Meta-Chart,“ [Võrgumaterjal]. Available: <https://www.meta-chart.com/>. [Kasutatud 26 veebruar 2015].
- [52] www.randombugs.com, „MySQL vs PostgreSQL Benchmarks,“ www.randombugs.com, [Võrgumaterjal]. Available: <http://www.randombugs.com/linux/mysql-postgresql-benchmarks.html>. [Kasutatud 4 mai 2015].

Lisa 1

<https://chriskokka@bitbucket.org/chriskokka/loputoo.git>