VILNIAUS UNIVERSITETAS

EKONOMIKOS IR VERSLO ADMINISTRAVIMO FAKULTETAS

JONAS JACIKEVČIUS

II kursas, verslo informacinės sistemos

Kursinis darbas

DIDŽIŲJŲ DUOMENŲ APDOROJIMO TECHNOLOGIJOS

Darbo vadovas –

Dr. Michail Kazimianec

Vilnius, 2022

Turinys

1. ĮVADAS

Duomenys buvo pradėti skaitmenizuoti 9 dešimtmečio pabaigoje, kai visas pasaulis dar buvo analoginis. Tuomet duomenų kiekiai buvo maži, o naujų duomenų generacija buvo lėta. Visi duomenys būdavo dokumentai, sudėti į eilutes ir stulpelius. Duomenų laikymas ir apdorojimas nekėlė problemų. Slenkant metams internetas pradėjo sparčiai populiarėti, kas lėmė duomenų kiekio sprogimą. Pasipylę duomenys turėjo įvairiausius formatus ir formas, o jų gausėjo kiekvieną sekundę. Pusiau struktūruoti ir visai nestruktūruoti duomenys turėjo įvairiausias formas: elektroniniai laiškai, vaizdo įrašai, nuotraukos, garso įrašai ir begalė kitų. Visi šie duomenys 2005 m. buvo pradėti vadinti didžiaisiais duomenimis (Big Data) (Phillips, 2018). Pradžioje žmonės bandė naudoti seną kompiuterinę įrangą šiems duomenims apdoroti, tačiau greitai tapo aišku, kad procesoriai per lėti, o kietųjų diskų talpa maža. 2006 metais pasirodžiusi Big Data apdorojimo technologijų programinė įranga „Apache Hadoop“ sukėlė perversmą. Vietoje brangių didelės talpos kietųjų diskų, buvo galima naudoti HDFS failų sistemą, kuri išskirsto failus į mažesnio dydžio blokus, kuriuos galime laikyti skirtingų talpų ir greičių kietuosiuose diskuose. MapReduce technologija leido duomenis išskaidyti į fragmentus, kad juos būtų galima apdoroti keliais skirtingais procesoriais vienu metu. Po išskaidyto apdorojimo duomenys vėl surenkami į vieną vietą. Dideli duomenys gali būti apibrėžti kaip didelės apimties, didelio greičio ir didelės įvairovės, kuriems būtinas naujas didelio našumo apdorojimas. Didelių duomenų tvarkymas yra sudėtinga ir daug laiko reikalaujanti užduotis, kuriai reikalinga galinga skaičiavimo infrastruktūra, kad būtų užtikrintas sėkmingas duomenų apdorojimas ir analizė. Šiame darbe apžvelgiama duomenų apdorojimo metodika, pateikiamas duomenų apdorojimo metodų apibrėžimas, charakteristikos ir skirstymas į kategorijas. Taip pat nagrinėjamas ryšys tarp didelių duomenų ir duomenų apdorojimo metodų ir didelių duomenų technologijų grupėse, įskaitant naujausių technologijų apžvalgą. Be to, bus aptariami mokslinių tyrimų iššūkiai, daugiausia dėmesio skiriant įvairių didelių duomenų sistemų, pavyzdžiui, Hadoop ir Spark, raidai ir skiriamas dėmesys duomenų apdorojimo metodų ir taikomųjų programų grupėms.

1.1 Darbo aktualumas

Organizacijos, kurios naudoja didžiuosius duomenis savo sistemose, gali pralenkti savo konkurentus rinkos ir marketingo analizėje, geriau personalizuoti reklamas ir sklandžiau aptarnauti savo klientus. Efektyvus didžiųjų duomenų naudojimas leidžia pirmauti rinkoje prieš konkurentus dėl efektyvesnio ir greitesnio verslo sprendimų priėmimo. Didieji duomenys yra sutinkami visose didžiųjų korporacijų srityse: energetikos industrijoje didieji duomenys leidžia naftos ir dujų tiekėjams atrasti žaliavų šaltinius; finansinių paslaugų sektoriaus įmonės gali

didžiuosius duomenis naudoti rinkos analizei ir rizikų valdymui; logistikos įmonės didžiųjų duomenų pagalba prižiūri tiekimo grandines ir optimizuoja maršrutus; valstybinės organizacijos gali didžiuosius duomenis naudoti masiniam gyventojų sekimui ir skubiųjų tarnybų veiklos gerinimui (Botelho, 2022).

1.2 Darbo problema

Sparčiai didėjant duomenų kiekiui ir įvairovei, būtina efektyviai ir optimaliai apdoroti didžiuosius duomenis, norint pasiekti geriausius rezultatus organizacijoje, tačiau kaip šiuos duomenis surinkti ir kokias technologijas taikyti juos apdorojant?

1.3 Darbo uždaviniai

- Paaiškinti kas yra didieji duomenys
- Nurodyti didžiųjų duomenų apdorojimo žingsnius

1.4 Darbo metodai

- Informacija buvo renkama iš mokslinių šaltinių ir straipsnių
- Programinės įrangos naudojimosi instrukcijos buvo surinktos iš programinės įrangos dokumentacijos

2. APIE BIG DATA

Duomenys nėra naudingi jokiai įstaigai, kol jie nėra tinkamai apdoroti. Duomenų apdorojimas yra būdas duomenis paversti į naudingą informaciją. Šis procesas yra būtinas verslų ir įstaigų strategijų kūrimui ir tobulinimui, rezultatų gerinimui. Šį procesą sudaro 4 žingsniai:

1. Duomenų surinkimas – pirmas ir būtinas žingsnis visuose duomenų apdorojimo procesuose. Duomenys yra surenkami iš turimų šaltinių (*data lakes*). Yra svarbu, kad turimi duomenys būtų patikimi, kad vėliau išgauta informacija būtų kokybiška.
2. Duomenų paruošimas – surinkti duomenys turi būti paruošti. Duomenys yra filtruojami, kad liktų tik kokybiški vienetai, tikrinama ar tarp jų nėra klaidų. Šis žingsnis yra svarbus, kad surinkti ir vėliau naudojami duomenys būtų kokybiški ir tarp jų nepasitaikytų duomenų su klaidomis ar neskaitomų duomenų.
3. Duomenų įvedimas – išvalyti ir paruošti duomenys turi būti pateikti apdorojimo programinei įrangai. Tai reiškia, kad duomenys turi būti suformatuoti atitinkamai, kad programinė įranga galėtų juos priimti ir juos apdoroti. Duomenų įvedimas yra pirmas žingsnis, kuriame duomenys pradeda virsti į naudingą informaciją.
4. Apdorojimas – šiame žingsnyje įvesti duomenys yra apdorojami tolimesnei interpretacijai. Apdorojimas vyksta naudojant įvairius algoritmus, mašininį mokymasį ir panašius procesus. Procesai priklauso nuo įvestų duomenų pobūdžio ir lūkesčių apdorotų duomenų analizei.

3. DUOMENŲ SURINKIMAS

Pirmasis duomenų apdorojimo etapas – duomenų surinkimas. Duomenys gali būti struktūruoti, nestruktūruoti ir pusiau struktūruoti. Struktūruoti duomenys turi griežtą formatavimą ir atitinka griežtus kriterijus. Struktūruotų duomenų pavyzdys – mokėjimo kortelių numeriai, GPS koordinatės. Nestruktūruoti duomenys yra duomenys laisva, neapibrėžta forma, ir gali būti skirtingų formatų kaip socialinės medijos įrašai arba elektroninių laiškų priedai Likę laiškai yra pusiau struktūruoti. Tai yra duomenys, kurių dalis turi struktūrą, o likusi – ne. Pavyzdžiui elektroniniai laiškai. Siuntėjas ir gavėjas turi elektroninio pašto adresą – struktūruoti duomenys, o laiško turinys nėra struktūruotas. Visas laiškas yra pusiau struktūruotas. Renkami duomenys priklauso šioms kategorijoms (Christiansen, 2022):

- **Tinklų duomenys -** šie duomenys yra renkami iš įvairių tinklų, įskaitant socialinius, informacinius tinklus, internetą, mobiliuosius tinklus ir pan.
- **Realaus laiko** arba tiesioginiai duomenys yra renkami iš transliavimo platformų kaip YouTube, Twitch, Skype, Netflix ir t.t.
- **Sandorių duomenys** yra renkami iš vartotojų, kai jie atlieka įvairius sandorius, kaip internetiniai atsiskaitymai, bankų perlaidos, apmokėjimai parduotuvės kasoje.
- **Geografiniai duomenys** yra renkami palydovais (nuotraukos ir vietos nustatymo). Į šiuos duomenys įeina gamtos procesai, orai, vandenynų veikla, žmonių judėjimas, transporto priemonių judėjimas, pastatai ir t.t.
- **Visuomenės sekimo duomenys** renkami specialiomis kameromis, fiksuojančiomis žmogaus eiseną.

Apibrėžę kategorijas duomenims, turime numatyti, kur juos surinksime:

- **Marketingo analitika** yra skaitmeninio marketingo varomoji jėga. Didžiausios elektroninės komercijos organizacijos, kaip „Amazon“, „IKEA“, „Walmart“ ir „Alibaba“ kasdien aptarnauja milijonus vartotojų ir turi susidoroti su milžiniškais duomenų kiekiais. Kiekvieno pirkimo metu, vartotojai turi įrašyti savo vardą, pavardę, elektroninio pašto adresą, fizinį adresą, mokėjimo būdą, telefono numerį. Kraštutiniais atvejais, vienas klientas gali užimti iki 2 gigabaitų duomenų (SalesForce pateikiama informacija). Šalia duomenų saugojimo ir naudojimo, verslai atlieka klientų analizes, taip siekdami gerinti jų apsipirkimo patirtį, geriau suprasti jų norus. Nors verslams tai

leidžia suprasti savo auditoriją ir gerinti pardavimus, tokie duomenų kiekiai nėra lengvai apdorojami ir reikalauja specialių technologijų bei įgūdžių.

- **Lojalumo ir nuolaidų kortelės** yra itin dažna praktika fizinėse parduotuvėse, kurios tikslas yra skatinti pirkėjų ištikimybę renkantis parduotuvių tinklą. Šios kortelės leidžia verslams sukurti kiekvieno kliento profilį su duomenimis, kurie leistų detaliai analizuoti kiekvieno kliento poreikius ir įpročius. Pavyzdžiui: kiek kartų klientas per savaitę apsipirkinėja, kokia jo vidutinė krepšelio suma, ar jis turi vaikų, jei turi – kiek. Šie klientų profiliai gali būti parduodami reklamų kompanijoms, kurios kurtų efektyvesnes reklamas, nutaikytas į tikslesnę auditoriją.

- **Palydovinės nuotraukos** yra efektyviausias būdas rinkti geografinius duomenis. Google priklausantys palydovai fotografuoja Žemę 50 – 70 kartų per dieną. Taip pat Google naudoja ir kitą metodą geografiniams duomenims rinkti – Google Street View. Po beveik viso pasaulio miestus važinėja automobiliai su pritaisytomis kameromis gebančiomis filmuoti viską 5 kadrų per sekundę greičiu 8k raiška, 360° horizontaliai ir 135° vertikaliai (Google pateikiama informacija).

- **Veikla socialinėse erdvėse.** Žmonės kasdien vidutiniškai socialiniuose tinkluose praleidžia 168 minutes (Macit, 2018). Socialiniai tinklai yra laikomi pagrindiniais šaltiniais dideliems duomenims. Nuotraukos, vaizdo įrašai, garso įrašai, tekstai. Nors ir naudotojai yra linkę dalintis duomenimis niekieno neprašomi, didžiųjų duomenų rinkimo, valdymo ir apdorojimo įrankiai yra būtini siekiant suvaldyti tokio masto duomenų srautus.

Žinodami, kur duomenis galime rasti, galime pradėti juos rinkti (Goworek, 2018):

- **Duomenų prašymas –** kurdami paskyras socialiniuose tinkluose ar elektroninėse parduotuvėse, privalome nurodyti asmeninę informaciją. Mažiausiai iš mūsų yra reikalaujama vartotojo vardo ir elektroninio pašto, tačiau gali reikėti įrašyti ir adresą, gimimo metus, įkelti nuotrauką.
- **Slapukai** (angl. *cookies*) yra populiariausias ir mažiausiai naudotojo pastangų reikalaujantis būdas duomenims surinkti. Iki 2011 metų svetainės net nebuvo įpareigotos pranešti naudotojams apie naudojamus slapukus.
- **Elektroninių laiškų sekimas** yra mažajai dalių naudotojų žinomas ir vis dar niekaip nekontroliuojamas duomenų rinkimo būdas. Jis vykdomas siunčiant klientams laiškus su prisegtu 1 pikselio dydžio paveikslėliu, kuris yra individualiame serveryje. Kai klientas atsidaro gautą elektroninį laišką, automatiškai yra užkraunami laiško priedai, įskaitant ir sekimo pikselį. Kai jis yra užkraunamas, pikselis pateikia užklausą į serverį, kuriame jis yra patalpintas. Taip siuntėjas gali žinoti ar gavėjas atidarė elektroninį laišką.
- **WebScraping (**duomenų ištraukimas) yra dar vienas duomenų didžiųjų duomenų surinkimo būdas, kuris gali būti visiškai automatizuotas. WebScraping yra būdas ištraukti bet kokius duomenis iš svetainių greitai su minimaliais žmogiškųjų išteklių resursais.

Prieš pradėdami rinkti duomenis, turime numatyto, koks bus duomenų tipas, iš kur rinksime duomenis ir kokius metodus jų surinkimui naudosime.

4. DUOMENŲ PARUOŠIMAS

Svarbiausias duomenų paruošimo procesas – duomenų valymas. Duomenų valymo metu klaidingi duomenys yra šalinami arba taisomi, tikrinami duomenų formatai ir jų klaidingumas. Esant dideliems duomenų kiekiams proporcingai auga rizika duomenų gadinimui ir klaidos faktorius įvesties metu. Jei duomenys nėra tikslūs, jų apdorojimas ir analizė nepavyks arba bus beverčiai. Nors duomenų valymui reikalingi skirtingi procesai ir sistemos dėl duomenų įvairovės, analizės lūkesčių ir organizacijos poreikių, šie esminiai žingsniai gali būti taikomi plačiai:

1. Dublikatų ir neaktualių duomenų šalinimas: rinkdami didžiuosius duomenis ilgainiui susidursime su įvestų duomenų dublikatais. Tai atsitinka dėl žmogiškos klaidos įvedant duomenis ar pakartotinų duomenų rinkimo tose pačiose vietose. Neaktualių duomenų naikinimas yra nemažiau svarbus žingsnis, reikalaujantis specialių algoritmų ir filtrų duomenims atrinkti. Pavyzdžiui, jei renkame duomenis apie kaimynų keliamą triukšmą (dažnumą, laiką, triukšmo lygį), duomenys, surinkti iš žmonių gyvenančių vienkiemiuose mums bus neaktualūs.

   Duomenų dublikatų naikinimas ir neaktualių duomenų šalinimas yra vieni plačiausių aspektų valant duomenis.
2. Struktūrinių klaidų taisymas: struktūrinės klaidos pasireiškia netiksliais failų pavadinimais, klaidomis duomenyse, rašybos klaidomis, taip pat sugadintais įrašais.
3. Trūkstamų duomenų tvarkymas: jei mūsų įrašuose trūksta reikalingų duomenų apdorojimui atlikti, apdorojimo algoritmai su tais duomenimis negalės daryti nieko. Tvarkydami trūkstamus duomenis turime 2 pasirinkimus: 1. Apskritai pašalinti ne iki galo užpildytus duomenų vienetus. 2. Suvesti trūkstamus duomenis, naudojant tendencijas, atpažįstančius algoritmus, mašininį mokymąsi ar naudodami *crowdsourcing* (savanoriškas sutelktines paslaugas).
4. Duomenų kokybės užtikrinimas: atsakę į visus šiuos klausimus teigiamai, galime tęsti duomenų apdorojimą: Ar duomenys logiški? Ar duomenys yra jiems priklausančiuose laukuose? Ar galime įžvelgti logišką tendenciją?

Duomenų valymo procesui naudojamos įvairios technologijos ir sistemos, leidžiančios atskirti kokybiškus duomenis ir juos struktūrizuoti.

„**Cleanix**“ sistemos prototipas

Wang [X] yra vienas iš duomenų valymo sistemos, pavadinimu „Cleanix“ autorių. „Cleanix“ sistema išskiria 4 problemines temas atskiriant kokybiškus duomenis nuo nekokybiškų (Wang, Li, Bung, Gao, Zhang, Li, 2015): nenormalios vertės ir reikšmės, ne iki galo įvesti duomenys, dublikatai, konfliktiniai duomenys (prieštaraujantys kitiems ar sau). „Cleanix“ sistema sprendžia šiuos nekokybiškų duomenų trūkumus. Sistema buvo kuriama taip, kad galėtų prisitaikyti prie duomenų kiekio ir taip nebūtų švaistomų kompiuterinių resursų. Programa buvo optimizuota, jog galėtų valyti duomenis ir pranešti apie jų kokybę vienu metu. Tai yra labai svarbu, jei kasdien reikia susidoroti su milžiniškais duomenų kiekiais. Visa sistema yra automatizuota ir atlieka 4 žingsnius: perskaito duomenis ir atskiria duomenis turinčius nenormalias reikšmes ir vertes; jei įmanoma ištaiso neteisingus duomenis; įrašo trūkstamus duomenis; naikina dublikatus ir taiso konfliktinius duomenis. Naudojami algoritmai ieško tendencijų tarp turimų duomenų ir taip užpildo trūkstamas spragas. Esminis sistemos trūkumas – didelis klaidos faktorius.

„SCARE“ sistema

Yakout, kartu su Berti-Equille, sukūrė SCARE (SCalable Automatic REpairing) duomenų valymo sistemą. Ši sistema yra pagrįsta mašininio mokymosi technologija. Sistemos struktūra (framework) prisitaiko prie duomenų apimties ir dalina duomenis į horizontalias pertvaras (partitions), kad užtikrintų sistemos lankstumą prisitaikant prie duomenų kiekio ir paralelinį duomenų apdorojimą. Sistemos tikslas yra naudojant mašininį mokymąsi užpildyti duomenų skyles pagal vyraujančias pateiktų duomenų tendencijas. Negana to, su šia sistema galima atkurti ir sugadintas duomenų bazes. Ši sistema savo funkcijas atlieka geriau su didesniais duomenų kiekiais. Kadangi jos veikimas yra pagrįstas mašininiu mokymusi, ji turi turėti medžiagos mokymuisi atlikti. Kuo mažesnis duomenų kiekis, tuo didesnė riziką netiksliam duomenų taisymui.

„KATARA“ sistema

„KATARA“ sistemos autoriai (Chu, Morcos, Ilyas, Ouzzani, Papotti, Tang, Ye) teigia, kad nei mašininis mokymasis, nei statistinės tendencijos, nei vientisumo paisymai negali užtikrinti duomenų tikslumo atkuriant sugadintus duomenis. „KATARA“ sistema yra paremta masine sistemos naudotojų įvestimi ir žinių bazių naudojimu. Pirmos dvi sistemos naudoja mašininį mokymąsi ir algoritmus duomenims valyti, tuo tarpu „KATARA“ kūrėjai, nepasitikėdami šiais duomenų valymo metodais, naudoja *crowdsourcing (*savanoriškos sutelktinės paslaugos*)*. Naudojantis šiuo metodu, nereikia remtis jau turimų duomenų tendencijomis, vietoj to galima prašyti žmonių įvesties. Nors tai ir didina tikslumą, tačiau reikalauja didelių žmogiškųjų resursų.

Duomenų valymo metodika priklauso nuo turimo duomenų kiekio, pobūdžio ir prieinamų resursų, todėl galima teigti, kad nėra vieno universalaus sprendimo duomenų valymui atlikti.

5. DUOMENŲ ĮVEDIMAS

Duomenų įvedimas yra būtinas žingsnis norint, kad mūsų struktūruoti ar nestruktūruoti duomenys patektų į didžiųjų duomenų apdorojimo failų sistemas.

Hadoop Distributed File System (Hadoop išskirstytų failų sistema) yra failų sistema, leidžianti talpinti įvairios apimties failus naudojantis kelių kompiuterių duomenų laikmenomis. Į HDFS failų sistemą įkelti failai yra dalinami į blokus, o blokai yra išdalinami kompiuterių tinklui. Pavyzdžiui: turime tekstinį struktūruotą duomenų failą vardai.txt, užimantį 150 megabaitų vietos. Kai failas yra įkeliamas į HDFS, jis yra išskirstomas į blokus (fragmentus). Kiekvienas blokas yra ganėtinai didelis. Numatytas bloko dydis yra 64 megabaitai. Kiekvienam blokui yra suteikiamas unikalus pavadinimas BLK ir priskiriamas numeris. Mūsų atveju failus vadinsime blk_1, blk_2 ir taip toliau. Pirmasis blokas yra 64 megabaitų dydžio, antrasis blokas taip pat yra 64 megabaitų dydžio, trečiajam blokui lieka 22 megabaitai. Kai failas yra įkeliamas į HDFS, kiekvienas blokas yra talpinamas į atskirą mazgą klasteryje. Turime žinoti, kuris blokas sudaro kurią failo dalį. Tam naudosime kitą kompiuterį, kuriame veikia NameNode sistema. Duomenys laikomi NameNode yra blokų pavadinimai. Šie duomenys yra vadinami metadata (meta duomenys). Juose yra nurodyta kur galima rasti failą ir kur galima rasti kitas 2 bloko atsargines kopijas. Visa ši sistema optimizuoja mūsų failo saugojimą ir prieigą. Svarbu prabrėžti, kad vienu metu gali būti tik vienas NameNode, tačiau patartina turėti bent kelias jo atsargines kopijas. Jeigu NameNode nebebus pasiekiamas, visi DataNodes nebeturės jokios prasmės.

5.1 HDFS

„Hadoop“ programinė įranga naudoja HDFS Pavyzdžiui naudosime vardai.txt struktūruotą tekstinį duomenų failą, kuriame pateikiamas lietuviškų vardų sąrašas. Prieš pradedant darbą su HDFS būtina ją suformatuoti. Demonstracijai naudojama Ubuntu 16.0.4.7 operacinė sistema; OPENJDK 8 Java platformos implementacija, OpenSSH ir Hadoop 2.3.7 aplinka, o komandos yra paremtos Hadoop dokumentacija.

```
$ hadoop namenode -format
```

Po formatavimo paleidžiame išskirstytą failų sistemą:

```
$ start-dfs.sh
```

Kai HDFS yra užkrauta, galime apžvelgti failų sąrašą mūsų HDFS kataloge:

```
$HADOOP_HOME/bin/hadoop fs -ls
```

Pradedame įkėlinėti duomenis į HDFS. Turime sukurti įvesties katalogą:

```
$HADOOP_HOME/bin/hadoop fs -mkdir /user/input
```

Perkeliame failus iš vietinės talpyklos į HDFS su komanda:

```
$HADOOP_HOME/bin/hadoop fs -put /home/vardai.txt /user/input
```

Pasitikriname ar failai persikėlė apžvelgdami jų sąrašą:

```
$HADOOP_HOME/bin/hadoop fs -ls /user/input
```

Jei mūsų failas vardai.txt yra sąraše, mūsų duomenys sėkmingai pateko į „Hadoop“ failų sistemą, kurioje vyks tolimesnis apdorojimas.

Duomenų įvedimas priklauso nuo naudojamos failų sistemos, todėl svarbu vadovautis programinės įrangos kūrėjų pateikta dokumentacija.

6. DUOMENŲ APDOROJIMAS

Duomenys gali būti apdorojami įvairia programine įranga. Bus apžvelgtos Hadoop ir Spark duomenų apdorojimo sistemos.

Hadoop struktūra susideda iš 3 sluoksnių:

1 Lentelė Hadoop struktūra

Hadoop HDFS	Duomenų laikymo išskirstyta failų sistema, leidžianti laikyti duomenis skirtinguose kompiuteriuose, taip užtikrinant optimalų duomenų apdorojimą ir mažesnę riziką jų netekti.
Hadoop YARN	Optimaliam resursų valdymui ir apdorojimo vykdymui naudojamas YARN komponentas. Jis susideda iš resursų valdytojo ir mazgų valdytojo.
MapReduce	Sistema išskaidanti duomenis į atskirus mazgus greitesniam jų apdorojimui.

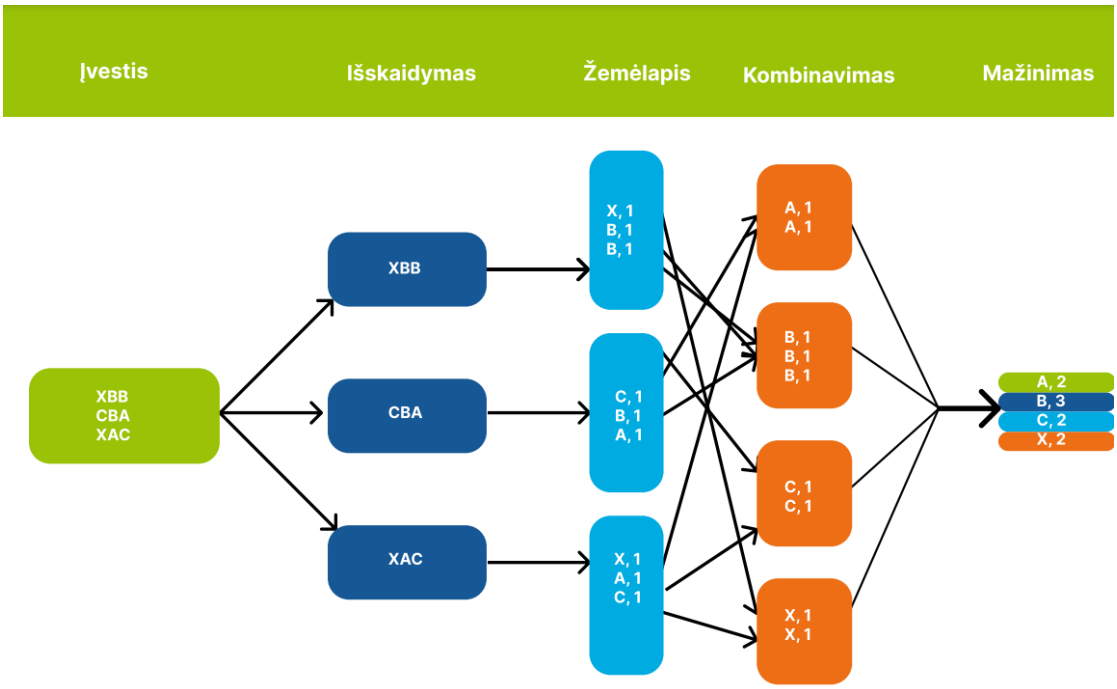
6.1 MapReduce

MapReduce yra sistema, kuri išskaido duomenis į mazgus, atskiria juos pagal tendencijas ir priskiria indikatorius, sukombinuoja duomenis ir galiausiai sumažina juos. MapReduce sistema buvo sukurta 2004 m., kompanijos Google. Databricks išleido tiriamąjį darbą, pristatantį šios sistemos prototipą „MapReduce: Simplified Data Processing on Large Clusters“. Jame buvo aprašyta MapReduce technologija, kuri buvo įkvėpta mažinimo funkcijų, naudojamų programavime. Tuomet MapReduce naudojo GFS (Google Fyle System) failų sistemą, kurios architektūra vėliau buvo naudojama ir HDFS. Google 2015 metais nustojo naudoti MapReduce sistemą dėl jos kompleksiškumo bei didelių resursų reikalavimo ir didžiąją dalį duomenų perkėlė į debesų kompiuteriją. Nors MapReduce technologija vis dar naudojama beveik visose organizacijose, kurios susiduria su didžiųjų duomenų apdorojimu, duomenų mokslininkai David J. DeWitt ir Michael Stonebraker teigia, jog MapReduce technologija yra jau pasenusi, dėl prastos naudotojų sąsajos su duomenų baze, prastos implementacijos, funkcionalumo trūkumo ir

nesuderinamumo su DBMS SQL įrankiais. DeWitt ir Stonebraker nėra vieninteliai duomenų mokslininkai, peikiantys šį duomenų apdorojimo metodą duomenis skaidant, tačiau alternatyvų nesiūlo.

Kad MapReduce sistemos veikimas būtų aiškesnis, bus pateiktas praktinis pavyzdys: įsivaizduokime, jog turime Lietuvos gyventojų vardų sąrašą. Tariamai turime 2,7 milijono įrašų, kurių apdorojimas truktų ilgai ir reikalautų didelių resursų, jei siektume tai atlikti įprastais apdorojimo metodais. MapReduce įvesties procesas vyksta 2 etapais: į įvestį turime įrašyti duomenų raktus ir vertes. Numatytosiose reikšmėse raktas yra duomenų eilutės numeris, o vertė yra lauko turinys. Tuomet ši raktų ir verčių pora yra įkeliama į žemėlapį, kuriame MapReduce sistema sugeneruoja naujai išvestą rakto ir vertės porą kiekvienai įvesties porai. Tuomet seka kombinavimo fazė, kurioje pradedame kombinuoti poras pagal mūsų norimą kriterijų. Mūsų atveju turime didelį vardų sąrašą, kurį galime sukombinuoti į 32 atskirtis pagal lietuviškos abėcėlės raides ir kiekvienos raidės atskirtyje sistema suras kiek vardų kartojasi toje atskirtyje. Pavyzdžiui: atskirtis A: Aida 5231 įrašai, Aidas 6985 įrašai [...]; atskirtis B: Balandis 12 įrašų, Baltė 7 įrašai ir taip iki paskutinio vardo. Galiausiai turime sukombinuotus (sumažintus) duomenis.

1 paveikslas. MapReduce veikimas



6.2 Hadoop YARN

YARN – Yet Another Resource Negotiator (dar vienas resursų valdytojas) (Hadoop YARN dokumentacija). YARN susideda iš 2 esminių komponentų: resursų valdytojo ir mazgų

valdytojo. Resursų valdytojas skirsto resursus tarp dirbančiųjų kompiuterių: procesorių darbą, operatyvinės atminties resursus, duomenų laikmenų užimtumą ir interneto greitį tarp kompiuterių. Mazgų valdytojas atsako už mazgų veikimą ir praneša resursų valdytojui apie jų darbą. Sistemos naudotojas pateikia užklausas, kurias vykdys mazgų valdytojas. Tuomet mazgų valdytojas patikrina sistemą ir sužino, su kokiais resursais darbas bus atliekamas. Tuomet resursų informacija yra išsiunčiama resursų valdytojui, kuris bus atsakingas už jų paskirstymą ir HDFS esančių failų apdorojimas yra pradedamas.

6.3 Spark

„Apache Spark“ yra naujesnė Hadoop kūrėjų didžiųjų duomenų apdorojimo platforma, pasirodžiusi 2009 m. UC Berkeley's AMPLab technologijų laboratorijoje ir paviešinta atviru kodu 2010 m. 2013 m. projektas buvo perduotas Apache Software Foundation fondui ir 2014 m. tai tapo pagrindiu Apache projektu. Apache „Spark“ pasiekė ne vieną pasaulio rekordą: 2014 m. apdorojo 100 terbaitų duomenų per 23 minutes, o 2016 m. apdorojo duomenis pigiausiai – 1 terabaitą duomenų apdorojo už 1.44$. „Spark“ yra Hadoop MapReduce patobulinimas, pagreitinantis MapReduce iki 100 kartų ir turi geriau optimizuotą blokų atkūrimą klaidos atveju (Amplab pateikta informacija). Esminiai skirtumai tarp „Spark“ ir MapReducer(IBM tinklaraštis, 2021):

- Hadoop duomenis tvarko kietuosiuose diskuose, tuo tarpu „Spark“ – operatyviojoje atmintyje. Dėl šios priežasties „Spark“ duomenų judėjimas yra daug greitesnis, nes programa nėra priklausoma nuo kietojo disko skaitymo ir įrašymų greičių, o kliaujasi daug greitesnės operatyviosios atminties resursais.
- Operatyviosios atminties naudojimas turi ir kitą pusę – didesnė kaina. Kadangi beveik visos „Spark“ operacijos vyksta operatyviojoje atmintyje, kuri yra daug brangesnė, operacijų kaina stipriai išauga. Lyginant kietųjų diskų ir operatyviosios atminties didmenines kainas, tokios pat talpos kietasis diskas kainuos maždaug 3-6 kartus pigiau nei operatyviosios atminties talpykla.
- Dar vienas „Spark“ pranašumas – mašininio mokymosi galimybės. Į „Spark“ yra įdiegta MLib biblioteka ir įvairūs įrankiai, galintys atlikti regresijas, klasifikacijas, vertinimus ir taip toliau.

Nors ir Spark daugeliu atveju yra daug pranašesnė sistema nei Hadoop, kiekviena šių sistemų turi pritaikymo atvejus, kuomet viena sistema yra naudingesnė už kitą:

2 Lentelė. Spark ir Hadoop naudojimo paskirtys

„Spark“	Hadoop
Apdorojant duomenis, kai rezultato reikia labai greitai	Duomenis reikia apdoroti išnaudojant kietųjų diskų skaitymo ir rašymo klaidas
Apdorojant duomenis realiu laiku	Apdorojant duomenis ribojant biudžetą
Atliekant daug paralelinių apdorojimų vienu metu	Atliekant neskubias duomenų apdorojimo užduotis
Apdorojant duomenis naudojantis mašininiu mokymusi	Apdorojant archyvus ir analizuojant istorinius duomenis

Didžiųjų duomenų apdorojimas vyksta sklandžiai, jei šiai procedūrai pasirenkame tinkamas sistemas ir turime pakankamai resursų duomenims apdoroti.

7. DUOMENŲ APDOROJIMO RIBOS

Eisenos atpažinimas

2018 m. buvo pradėtas tyrinėti GKI (*Gait Kintetic Index* (eisenos kinetinis indeksas)), kuris leidžia identifikuoti žmogų pagal jo eisenos metu sukuriama kinetinę energiją. Indeksas remiasi 6 parametrais (Cimolin, Condoluci, Costici, Galli, 2018):

- Klubų pagreičiu
- Kelių pagreičiu
- Kulkšnių pagreičiu
- Klubų jėga
- Kelių jėga
- Kulkšnių jėga

Šie duomenys yra laikomi biometriniais, kadangi kiekvienas žmogus skirtingai judina savo kūno masę ir yra pranašesni už kitus biometrinius duomenis, kaip piršto antspaudų atpažinimą, veido atpažinimą ar rainelės atpažinimą, kadangi tirti eiseną galima dideliu atstumu nuo tiriamųjų, tiriamieji gali būti atpažinti be jų sutikimo ir kooperavimo ir eisenos nuslėpti neįmanoma, kaip slepiami pirštų antspaudai pirštinėmis ar veidai kaukėmis.

Šiam procesui yra reikalinga bent 1 vaizdo kamera. Kamera yra prijungiama prie kompiuterio, kuris gali apdoroti kameros pateikiamą ir informaciją ir atskirti skirtingas tiriamas kūno dalis. Tuomet yra daromi tų kūno dalių kadrai ir apdorojami pagal algoritmus, kurie nustato klubų, kelių ir kulkšnių sukuriamos kinetinės energijos kiekį ir jėgą bei apskaičiuoja individualų indeksą. Lyginant eisenos atpažinimą su kitų biometrinių duomenų rinkimu, kaip piršto antspaudų atpažinimu, eisenos atpažinimui reikalingas didelių duomenų apdorojimas, kadangi juos apdoroti reikia greitai, duomenų kiekis yra didelis ir duomenys yra skirtingų kategorijų, o dalis jų nėra struktūrizuoti.

Nors eisenos atpažinimo technologija yra tobulinama ir vis dar negali būti naudojama plačiai, yra kuriama sistema eisenos atpažinimui per neskaidrius objektus kaip sienas ar žmonių minias. Tam yra naudojama WiFi technologija. WiFi veikia radijo bangomis, kurios įprastai 2.4 GHz arba 5 GHz dažniais sukuria ryšį tarp maršrutizatoriaus, kuris skleidžia bangas ir įrenginių turinčias antenas, kurios gali šį ryšį priimti (Chruszczyk, Zając, Adam, Grzechca, Damian, 2016).

Trumpesnio dažnio bevielio ryšio bangos patiria mažiau nuostolių įveikdamos kliūtis, tačiau gali pernešti mažesnį duomenų kiekį, tuo tarpu ilgesnio dažnio bangos perneša duomenis greičiau, tačiau fizines kliūtis ir didesnius atstumus įveikia sunkiai. Manipuliuojant bangų dažniais ir fiksuojant ryšio delsą tarp maršrutizatoriaus ir įrenginio priimančio bangas, galima grafiškai atvaizduoti su kokiomis kliūtimis susiduria bangos, kol pasiekia savo tikslą. Atliekant tai daug kartų per sekundę, galime filmuoti kliūtis ir jų judesį už neskaidrių objektų.

Numatoma, kad ateityje šios technologijos susijungs ir bus galima stebėti ir atpažinti žmonių eiseną už sienų. Vienas pagrindinių faktorių, stabdančių šių technologijų tobulėjimą masiniu lygmeniu – duomenų apdorojimo sparta. Apjungiant šias dvi technologijas, reikia apdoroti ir analizuoti milžiniškus duomenų kiekius vienu metu. Tam reikia ne tik itin galingos techninės įrangos, bet ir pažangesnės technologijos duomenims apdoroti, nei turima dabar. Rinkos lyderiai Hadoop ir Spark nėra optimizuoti vienu metu apdoroti vizualius ir tekstinius duomenis bei sieti juos ryšiais tokiu dideliu mastu. Dėl šios priežasties yra išnaudojamas neoptimalus resursų kiekis duomenų apdorojimo metu.

IŠVADOS

Išsiaiškinę didžiųjų duomenų sąvoką ir apžvelgę jų apdorojimo metodus, susipažinome su didžiųjų duomenų tipais ir rinkimo metodikomis. Išsiaiškinę, kaip duomenis gauti, sužinojome, kaip tuos duomenis paruošti apdorojimui ir suvesti į pasirinktą sistemą. Galiausiai buvo pristatytos ir palygintos duomenų apdorojimo technologijos bei parodytas didžiųjų duomenų apdorojimo trūkumas apdorojant ir susiejant didelio masto skirtingo formato duomenis.

ŠALTINIAI

Amplab pateikta informacija https://amplab.cs.berkeley.edu/category/spark-2/

Andres Phillips, A history and timeline of big data (2018) https://www.techtarget.com/whatis/feature/A-history-and-timeline-of-big-data

Bridget Botelho Big Data (2022) https://www.techtarget.com/searchdatamanagement/definition/big-data

Chruszczyk, Lukas & Zając, Adam & Grzechca, Damian. (2016). Comparison of 2.4 and 5 GHz WLAN Network for Purpose of Indoor and Outdoor Location. International Journal of Electronics and Telecommunications. 62. 10.1515/eletel-2016-0010.

Cimolin, V., Condoluci, C., Costici, P. F., & Galli, M. (2018). A proposal for a kinetic summary measure: the Gait Kinetic Index. Computer Methods in Biomechanics and Biomedical Engineering, 1-6. https://doi.org/10.1080/10255842.2018.1536750

Databricks pateikta informacija https://databricks.com/glossary/mapreduce#:~:text=History%20of%20MapReduce,commonly%20used%20in%20functional%20programming.

DeWitt ir Michael Stonebraker Relational Database Experts Jump The MapReduce Shark https://typicalprogrammer.com/relational-database-experts-jump-the-mapreduce-shark

Google pateikiama informacija https://developers.google.com/streetview/ready/specs-prograde

Hadoop dokumentacija https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Hadoop dokumentacija https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YarnCommands.html

Hongzhi Wang, Mingda Li, Yingyi Bu, Jianzhong Li, Hong Gao ir Jiacheng Zhang Cleanix: a Parallel Big Data Cleaning System (2015) https://sigmodrecord.org/publications/sigmodRecord/1512/pdfs/06_systems_Wang.pdf

Hüseyin Bilal Macit ir Gamze Macit Affects of Social Media Addiction, A Survey (2018) https://www.researchgate.net/publication/330162011_Affects_of_Social_Media_Addiction_A_Survey

IBM tinklaraštis (2021) https://www.ibm.com/cloud/blog/hadoop-vs-spark#:~:text=Spark%20is%20a%20Hadoop%20enhancement,to%20100x%20faster%20than%20MapReduce.

Krzysztof Goworek Best data collection methods for improving your customers base https://tasil.com/insights/data-collection-methods/

Lauren Christiansen, How To Collect Data (2021) https://zipreporting.com/en/data-analysis-method/how-to-collect-data.html

Mohamed Yakout, Laure Berti-Equille ir Ahmed K. Elmagarmid Don’t be SCAREd: Use SCalable Automatic REpairing withMaximal Likelihood and Bounded Changes (2013) https://www.researchgate.net/publication/262218892_Don't_be_SCAREd_Use_SCalable_Automatic_REpairing_with_maximal_likelihood_and_bounded_changes

SalesForce pateikiama informacija https://help.salesforce.com/s/articleView?id=sf.overview_storage.htm&type=5

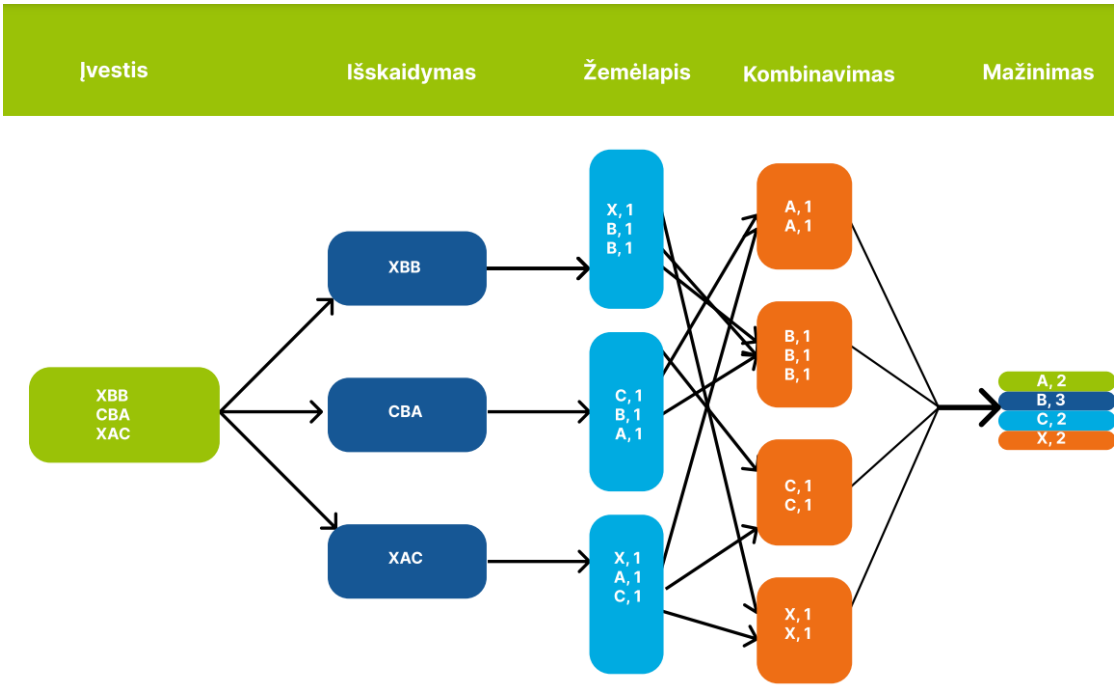
Xu Chu, John Morcos, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Nan Tang ir Yin Ye KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing (2016) https://cs.uwaterloo.ca/~ilyas/papers/ChuSIGMOD2015.pdf

PRIEDAI

1 Lentelė Hadoop struktūra

Hadoop HDFS	Duomenų laikymo išskirstyta failų sistema, leidžianti laikyti duomenis skirtinguose kompiuteriuose, taip užtikrinant optimalų duomenų apdorojimą ir mažesnę riziką jų netekti.
Hadoop YARN	Optimaliam resursų valdymui ir apdorojimo vykdymui naudojamas YARN komponentas. Jis susideda iš resursų valdytojo ir mazgų valdytojo.
MapReduce	Sistema išskaidanti duomenis į atskirus mazgus greitesniam jų apdorojimui.

1 paveikslas. MapReduce veikimas



z2 Lentelė. Spark ir Hadoop naudojimo paskirtys

„Spark“	Hadoop
Apdorojant duomenis, kai rezultato reikia labai greitai	Duomenis reikia apdoroti išnaudojant kietųjų diskų skaitymo ir rašymo klaidas
Apdorojant duomenis realiu laiku	Apdorojant duomenis ribojant biudžetą
Atliekant daug paralelinių apdorojimų vienu metu	Atliekant neskubias duomenų apdorojimo užduotis
Apdorojant duomenis naudojantis mašininiu mokymusi	Apdorojant archyvus ir analizuojant istorinius duomenis