

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

**Rodiklių duomenų kaupimas, transformavimas ir
analizė, naudojant NoSQL duomenų bazę**

**(Storage, transformation and analysis of indicator data with the
help of NoSQL database)**

Kursinis darbas

Atliko:	3 kurso 5 grupės studentas Vytautas Žilinas	(parašas)
Darbo vadovas:	lekt. Andrius Adamonis	(parašas)

Vilnius – 2018

TURINYS

IVADAS	3
1. RODIKLIŲ DUOMENYS	4
1.1. Apibrėžimas	4
1.2. Charakteristikos	4
2. RODIKLIŲ DUOMENŲ APDOROJIMO TIPAI	6
2.1. Reliacinės duomenų bazės duomenų apdorojimas	6
2.2. Srautinis duomenų apdorojimas	6
2.3. Kiti duomenų apdorojimo tipai	6
2.3.1. Paketinis duomenų apdorojimas	6
2.3.2. Lambda architektūra	6
2.3.3. Kappa architektūra	6
3. TESTINIŲ DUOMENŲ GENERATORIUS	7
3.1. Apibūdinimas	7
3.2. Paskirtis ir panaudojimo būdas	7
4. SPRENDIMO NAUDOJANČIO RELACINĘ DUOMENŲ BAZĘ PRALAIIDUMO TESTAS	8
4.1. Apibūdinimas	8
4.2. Testavimo rezultatai	8
5. SRAUTINIO APDOROJIMO ARCHITAKTŪROS	9
5.1. „Apache Storm”	9
5.2. „Apache Spark”	9
5.3. „Apache Flink”	9
6. SRAUTINĖS ARCHITEKTŪROS SPRENDIMO PRALAIIDUMO TESTAS	10
6.1. Apibūdinimas	10
6.2. Rezultatas	10
7. EXPERIMENTO APIBENDRINIMAS	11
7.1. Rezultatai	11
7.2. Eksperimento išvados	11
REZULTATAI IR IŠVADOS	12
LITERATŪRA	13

Įvadas

Darbo tikslas: Eksperimento būdu išbandyti rodiklių duomenų kaupimo, transformavimo ir analizės sprendimus, palyginant sprendimą, naudojanti reliacinę duomenų bazę, su sprendimu naudojančiu srautinį duomenų apdorojimą.

Užduotys:

1. Analizės būdu palyginti srautinį ir reliacinį duomenų apdorojimo sprendimą.
2. Sukurti testiniu duomenų generatorių.
3. Išmatuoti reliacinės duomenų bazės sprendimo pralaidumą.
4. Atlikti skirtingų srautinio duomenų apdorojimo sprendimo architektūrų analizę, ir pasirinkti vieną iš jų tyrimui.
5. Išmatuoti pasirinktos srautinio duomenų apdorojimo architektūros sprendimo pralaidumą.

1. Rodiklių duomenys

1.1. Apibrėžimas

Rodiklių duomenys - tai didelių duomenų tipas, kurį galima transformuoti ir analizuoti ir kuris yra sugrupuotas pagal rodiklius, pavyzdžiui: bazinė mėnesio alga, mirusiųjų skaičius pagal mirties priežastis, krituliai per metus. Šie duomenys dažniausiai yra saugomi reliacinėse duomenų bazėse, kur užklausus vartotojui skaičiuojami apibendrinti rodikliai - sumos, vidurkiai ir kita statistika. Lietuvoje pagrindinis rodiklių duomenų bazės pavyzdys yra „Lietuvos statistikos departamento“ duomenų bazė, kurios duomenis galima pasiekti <https://osp.stat.gov.lt/statistiniu-rodikliu-analize#/> puslapyje, kuris leidžia ieškoti duomenis pagal vieną arba kelis rodiklius. Didesnis pavyzdys yra „DataBank“ <http://databank.worldbank.org> - pasaulinio lygio rodiklių duomenų bazių rinkinys, turintis 69 skirtingas duomenų bazes, pavyzdžiui - „World development indicators“, „Gender statistics“ ir kitus[Ban18].

1.2. Charakteristikos

Apibrėžime minėjau, kad rodiklių duomenis yra didelių duomenų tipas, todėl galime jiems pritaikyti didelių duomenų charakteristikas ir apsibrėžti, kurios iš jų mums sudaro daugiausiai problemų. Šie iššūkiai apibrėžiami Gartner's Doug Laney pristatytu 3V modeliu[Lan01], kuris vėliau buvo papildytas Bernard Marr iki 5V modelio[Mar14]:

- Tūris (angl. Volume). Apibrėžia generuojamų duomenų kiekius. Didelių duomenų atveju yra šnekama apie duomenų kiekius, kuriuos yra sudetinga arba neįmanoma saugoti ir analizuoti tradicinėmis duomenų bazių technologijomis. Rodiklių duomenų kiekiai dažniausiai nesudaro problemos saugojant, tačiau didelė problema yra rodiklių duomenų analizė, kadangi tuos pačius duomenis reikia apdoroti pagal neapribotą skaičių skirtingų rodiklių.
- Greitis (angl. Velocity). Apibrėžia greitį, kuriuo nauji duomenis yra generuojami. Rodiklių duomenų atveju, tai yra labai svarbu, kadangi nauji duomenis, kurie gali tikti skirtingiems rodikliams yra generuojami visais laikais.
- Įvairovė (angl. Variety). Apibrėžia duomenų tipus. Duomenys gali būti: strukturizuoti, nestrukturizuoti arba dalinai strukturizuoti[ZE⁺11]. Rodiklių duomenis dažniau yra strukturizuoti, todėl tai nėra aktualus iššūkis.
- Tikrumas (angl. Veracity). Apibrėžia duomenų tesingumą ir kokybę. Pavyzdžiui, jeigu analizuotume „Twitter“ socialinio tinklo žinučių turinį gautume daug gramatikos klaidų, naujadarų, slengo. Statistinio departamento atveju duomenys visada bus tvarkingi, kadangi tai dažniausiai yra duomenys surinkti iš dokumentų ir apklausų, o ne laisvo įvedimo.
- Vertė (angl. Value). Apibrėžia duomenų ekonominę vertę. Rodiklių duomenys yra labai vertingi įstaigoms, nes dažniausiai tos įstaigos užsiema tik rodiklių duomenų kaupimu ir analizė, iš techninės pusės ši charakteristika yra svarbi iš tos pusės, kad duomenų apdorojimo ir kaupimo sprendimai labai stipriai daro įtaką įstaigos, kaupiančios rodiklių duomenis, ekonomikai. Taip pat šių duomenys ir jų analizė turi būti pasiekiami be prastovos laiko.

Pagal apibrėžtas charakteristikas matome, kad pagrindiniai rodiklių duomenų iššūkiai yra tūris, greitis ir vertė. Todėl mūsų bandomas sprendimas turi galėti greitai susidoroti su dideliu kiekių skirtingo pobūdžio duomenų ir turi sugebėti greitai atvaizduoti pokyčius atsiradus naujiems duomenims, taip pat turi būti įmanoma šį sprendimą paleisti į realią aplinką nepertraukiant įstaigos veiklą.

2. Rodiklių duomenų apdorojimo tipai

2.1. Reliacinės duomenų bazės duomenų apdorojimas

Aprašoma kaip generacija vykstu stored procedūros budu

2.2. Srautinis duomenų apdorojimas

Aprašoma kaip vykstu srautinis duomenų apdorojimas su schema.

2.3. Kiti duomenų apdorojimo tipai

2.3.1. Paketinis duomenų apdorojimas

Paketinis duomenų apdorojimas (angl. Batch processing) - didelių duomenų apdorojimo tipas, kai surinktas didelis duomenų kiekis nėra apdorojamas iš karto, o sukaupiamas ir vėliau apdorojamas visas iš karto. Mūsų reliacinės duomenų bazės apdorojimas naudoja primityvią paketinio duomenų apdorojimo versiją. Daug geriau yra žinomas kita paketinio duomenų apdorojimo architektūra - „Apache Hadoop”. Ši architektūra naudoja „Google” sukurtą „Map-Reduce” konceptą[DG08], kuris dideli duomenų kiekį suskaido į mažus rinkinius ir daro apdorojimą paraleliai ant visų rinkinių.[RHK⁺15] Ši architektūra mums netinka, nes mes norime rezultatą gauti tik įdėję naujų duomenų.

2.3.2. Lambda architektūra

Lambda architektūra - didelių duomenų apdorojimo architektūra naudojanti srautinio ir paketinio apdorojimo architektūrą. Su šia architektūra bandoma subalansuoti latenciją, pralaidumą ir klaidų toleranciją, naudojant paketinį apdorojimą tikslams ir išsamiesiems duomenų pakentams ir tuo pačiu metu naudoti srautinį apdorojimą greitai analizei[HKV14]. Tačiau ši architektūra turi vieną trūkumą - norint naudoti šią architektūrą reikia palaikyti dvi skirtingas architektūras, kad kodas būtų atnaujinamas abiejose architektūrose vienu metu[Kre14]. Relaus pavyzdys tokios architektūros būtų: Kafka žinučių sistema siunčianti duomenis į „Apache Storm”, kur duomenis apdorojami srautiškai, ir „Apache Hadoop”, kur duomenis apdorojami paketiškai, ir rezultatus saugant skirtingose duomenų bazės lentelėse.

2.3.3. Kappa architektūra

Kappa architektūra - tai supaprastinta lambda architektūra, kuri vietoj paketinio apdorojimo naudoja papildomą srautinio apdorojimo sistemą. Pavyzdžiui yra vienas srautinio apdorojimo darbas, kuris veikia visą laiką ir atvaizduoja duomenis gyvai, ir yra kitas darbas kuris paleidžiamas kas kažkiek laiko, kuris susirenka per visą tą laiką susikaupusius duomenis ir praleidžia per save srautu. Kadangi kodai yra vienodi, tai nebeatsiranda anksčiau minėtos problemos su skirtingų sistemų palaikymu[Kre14; Pat14].

3. Testinių duomenų generatorius

3.1. Apibūdinimas

Papasakoti čia apie Kafka

3.2. Paskirtis ir panaudojimo būdas

Papasakoti čia apie throtlinimo testavimą

4. Sprendimo naudojančio relaicinę duomenų bazę pralaidumo testas

4.1. Apibūdinimas

4.2. Testavimo rezultatai

5. Srautinio apdorojimo architektūros

Šiame skyriuje palyginsiu skirtingas srautinio apdorojimo architektūrų privalumus ir trūkumus ir pasirinksiu vieną su kuria sukursiu sprendimą rodiklių duomenų apdorojimui.

5.1. „Apache Storm”

5.2. „Apache Spark”

5.3. „Apache Flink”

6. Srautinēs architektūros sprendimo pralaidumo testas

6.1. Apibūdinimas

6.2. Rezultatas

7. Experimento apibendrinimas

7.1. Rezultatai

7.2. Eksperimento išvados

Rezultatai ir išvados

Darbo rezultatai:

- Išnagrinėti rodiklių duomenų analizės būdai pagal jų privalumus ir trūkumus.
- Sukurtas testinių duomenų generatorius, kurio pagalba buvo vykdomas pralaidumo testavimas.
- Atliktas pralaidumo testas Micorsoft SQL Express duomenų bazei su sukurtu testiniu duomenų generatorium ir tarpine Python aplikacija.
- Išanalizuoti skirtingos srautinio apdorojimo architektūros ir pasirinkta tinkamiausia sprendimo kurimui.
- Sukurtas sprendimas su pasirinkta srautinio apdorojimo architektūra.
- Atliktas pralaidumo testas sukurtam srautinio apdorojimo sprendimui.
- Palyginti sprendimų pralaidumo testų rezultatai.

Darbo išvados:

-
-

Literatūra

- [Ban18] The World Bank. List of databank databases. <http://databank.worldbank.org/data/databases/>, 2018.
- [DG08] Jeffrey Dean ir Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [HKV14] Zirije Hasani, Margita Kon-Popovska ir Goran Velinov. Lambda architecture for real time big data analytic. *ICT Innovations*, 2014.
- [Kre14] Jay Kreps. Questioning the lambda architecture. *Online article*, July, 2014.
- [Lan01] Doug Laney. 3d data management: controlling data volume, velocity and variety. *META Group Research Note*, 6(70), 2001.
- [Mar14] Bernard Marr. Big data: the 5 vs everyone must know. *LinkedIn Pulse*, 6, 2014.
- [Pat14] Milinda Pathirage. What is kappa architecture. <http://milinda.pathirage.org/kappa-architecture.com/>, 2014.
- [RHK⁺15] Shyam R, Barathi Ganesh Hullathy Balakrishnan, Sachin Kumar S, Prabaharan Poor-nachandran ir Soman Kp. Apache spark a big data analytics platform for smart grid. 21:171–178, 2015-12.
- [ZE⁺11] Paul Zikopoulos, Chris Eaton ir k.t. *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.