FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

BI-BIG

Semestrální práce

Autor:
Pavel Jahoda

Obsah

1 Úvod

Pro svoji závereˇcnou práci jsem se rozhodl použít reálná data charitativní organizace DonorsChoose.org, které podporuje veřejné vzdělání v USA. Data obsahují informace o dárcích, jednotlivé darované částky včetně dalších informacích o jednotlivých darech a v neposlední řadě obsahují stručné informace na co byli peníze využity. Nejprve jsem dělal agregace za použití Spark, poté nahrál nástrojem Logstash data do vyhledávacího nástroje Elasticsearch, kde jsem vytvořil index, který jsem využil na tvorbu grafů v Elasticsearch pluginu Kibana.

2 Data

Data, která jsem využil naleznete na adrese https://www.kaggle.com/donorschoose/io. Z šesti možných csv souborů jsem si vybral, že budu pracovat s csv soubory Donations, Donors a Schools.

2.1 Donors

Donors obsahuje data o dárcích. Soubor *Donors.csv* obsahuje okolo 2.1 milionů řádků. Data se skládají z 5 sloupců. Sloupec *Donor ID* obsahuje hexadimální číslo, které reprezentuje identifikátor dárce. *Donor City* a *Donor State* obsahuje text reprezentující název města dárce a stát z kterého daný dárce pochází. Povolené znaky jsou a-z, A-Z nebo mezera. Sloupec *Donor Is Teacher* zodpovídá jestli je dárce učitel, povolené hodnoty jsou pouze "Yes"nebo "No". Poslední sloupec *Donor Zip* obsahuje celá čísla 1-999 udávající první tři amerického ekvivalentu poštovního směrovacího čísla.

Donor ID	Donor City	Donor State	Donor Is Teacher	Donor Zip
00000ce845c00cbf0686c992fc369df4	Evanston	Illinois	No	602
00002783bc5d108510f3f9666c8b1edd	Appomattox	other	No	245
00002d44003ed46b066607c5455a999a	Winton	California	Yes	953
00002eb25d60a09c318efbd0797bffb5	Indianapolis	Indiana	No	462
0000300773fe015f870914b42528541b	Paterson	New Jersey	No	75
00004c31ce07c22148ee37acd0f814b9		other	No	
00004e32a448b4832e1b993500bf0731	Stamford	Connecticut	No	69
00004fa20a986e60a40262ba53d7edf1	Green Bay	Wisconsin	No	543
00005454366b6b914f9a8290f18f4aed	Argyle	New York	No	128
0000584b8cdaeaa6b3de82be509db839	Valparaiso	Indiana	No	463

Tabulka 1: Donors

2.2 Donations

Donations obsahuje data o jednotlivých darech. Soubor *Donations.csv* obsahuje okolo 4.7 milionů řádků. První tři sloupce zleva, neboli *Project ID*, *Donation ID* a *Donor ID* obsahují unikátní identifikátory projektu, daru a dárce. Povolené hodnoty jsou pouze čísla v hexadecimální podobě. Sloupec *Donation Included Optional Donation* obsahuje pouze hodnoty "No"a "Yes"a vyjadřuje jestli dárce dal 15% z darované částky charitativní organizaci DonorsChoose. *Donation Amount* je číslo vyjádřující darovanou částku. Darovaná částka může nabývat i desetinných hodnot. Sloupec *Donor Cart Sequence* obsahuje celá čísla a poslední sloupec *Donation Received Date* obsahuje datumy ve formátu "YYYY-MM-DD HH:MM:SS"a ukazují, kdy charita obdržela darovanou částku.

Project ID	Donation ID	Donor ID	Donation Included Optional Donation	Donation Amount	Donor Cart Sequence	Donation Received Date
00000989152(	68872912085(	1f4b5b6e684(	No	178.37	11	2016-08-23 13:15:57
00000989152(	dcf1071da3a(	4aaab6d244b	Yes	25	2	2016-06-06 20:05:23
00000989152(	18a234b9d1e	0b0765dc9c7(	Yes	20	3	2016-06-06 14:08:46
00000989152(	38d2744bf91(	377944ad61f(	Yes	25	1	2016-05-15 10:23:04
00000989152(	5a032791e31(	6d5b22d39e6	Yes	25	2	2016-05-17 01:23:38
00000989152(	8cea27f0cc03	896c75c9b8d(	Yes	15	1	2016-06-04 17:58:55
00000ce845c(	39af862cb04(	8a1875762c8(	Yes	50	1	2013-02-27 09:07:51
00000ce845c(	c47f78571f62	a3f070e439d(	Yes	50	2	2013-02-27 09:53:12
00000ce845c(	19351e1d9ae(	bd323208dc7	Yes	200	2	2013-02-17 21:36:24
00000ce845c(	d5364b1bb3(	6dd6113f89f2	Yes	10	44	2013-02-27 10:32:22

Tabulka 2: Donations

2.3 Schools

Schools obsahuje data o jednotlivých školách zapojených v projektu. Soubor *Schools.csv* obsahuje okolo 70 tisíc řádků. Soubor má 6 sloupců z nichž první je unikátní identifikátor v podobě hexadecimálního čísla. Sloupec *School Name* reprezentuje název školy. Přijatelné hodnoty jsou textové řetězce obsahující znaky a-z, A-Z nebo mezeru. *School Metro Type* popisuje jakého je typu z hlediska urbanismu. Povolené hodnoty jsou "rural", "suburban", "urban"nebo "unknown". Sloupec *School Percentage Free Lunch* udává procento žáků, kteří dostávají v rámci sociální podpory obědy zdarma. Povolené hodnoty jsou celá čísla od 0 do 100. Sloupec *School State* obsahuje textový řetězec, který říká ve kterém státě se škola nachází a sloupec *School Zip* obsahuje celá čísla 10000 až 99999 udávající americký ekvivalent poštovního směrovacího čísla.

School ID	School Name	School Metro Type	School Percentage Free Lunch	School State	School Zip
00003e0f	Capon Bridge Middle School	rural	56	West Virginia	26711
00004e32	The Woodlands College Park High School	urban	41	Texas	77384
0002021b	Samantha Smith Elementary School	suburban	2	Washing	98074
0004604f	Kingsbury Country Day School	unknown	76	Michigan	48370
0004c9d5	Redwater Elementary School	rural	50	Texas	75573
0004ffe3	Math & Science Success Academy	unknown	63	Arizona	85706
000622b5	Harbor Science & Arts Charter School	urban	17	New York	10029
000630ab	Spears Creek Child Development Center	unknown	15	South Carolina	29045
00064eac	Leadership Public School	urban	46	Californi	95122
00066582	Henking Primary School	suburban	29	Illinois	60025

Tabulka 3: School

3 Postup

3.1 Nahrání dat do Sparku

První úkol bylo vytvořit nový dataset, který bude agregovat data z jednoho původního datasetu. Tento bod, stejně jako další agregační úkoly jsem dělal ve Sparku. Nejprve jsem všechny 3 csv soubory nahrál do sparku. Postup byl následující.

```
cd BIG-FinalProject/spark
docker build -f spark.df -t spark .
docker-compose up -d
```

Data do sparku nahrajeme z Hadoop distribuovaného file systému HDFS, takže neprve je do HDFS musíme dostat a to následujícími příkazy v nové příkazové řadce.

```
docker run --name hadoop -v /home/pjahoda/FIT/BIG/BIG-FinalProject/
    ↪ logstash/datasets/:/tmp/datasets -t -i sequenceiq/hadoop-docker
    ↪  /etc/bootstrap.sh -bash
export PATH=$PATH:/usr/local/hadoop/bin/
hdfs dfs -mkdir /test
hdfs dfs -put ./tmp/datasets/Donors.csv /test/Donors.csv
hdfs dfs -put ./tmp/datasets/Donations.csv /test/Donations.csv
hdfs dfs -put ./tmp/datasets/Schools.csv /test/Schools.csv
```

Poté v původní příkazové řádce spustíme container, který reprezentuje driver program a následně v jeho příkazové řádce spustíme spark-shell připojený na master. V mém případě byla adresa masteru 172.17.0.2.

```
docker run -it -p 8088:8088 -p 8042:8042 -p 4041:4040 --name driver -h
    ↪  driver spark:latest bash
spark-shell --master spark://172.17.0.2:7077
```

Ve spark shellu nahrajeme data do proměnných se kterými budeme dále pracovat.

```
val donors = spark.sqlContext.read.format("csv").option("header", "
    ↪ true").option("inferSchema", "true").load("hdfs
    ↪ ://172.17.0.5:9000/test/Donors.csv")
val donations = spark.sqlContext.read.format("csv").option("header", "
    ↪ true").option("inferSchema", "true").load("hdfs
    ↪ ://172.17.0.5:9000/test/Donations.csv")
val schools = spark.sqlContext.read.format("csv").option("header", "
    ↪ true").option("inferSchema", "true").load("hdfs
    ↪ ://172.17.0.5:9000/test/Schools.csv")
```

3.2 Agregace na jednom datasetu

První úkol, agregovat data z jednoho původního datasetu, splníme následujícím příkazem.

```
val result = donors.groupBy("Donor City").count()
```

3.3 Agregace na dvou datasetech

Další úkol bylo vytvořit nový dataset, který bude agregovat data ze dvou původních datasetů najednou. Nejprve jsem udělal inner join nad dárcema a jejich darama.

```
val donor = donors.select(donors("Donor ID").as("id"),donors("Donor
    ↪ City"),donors("Donor State"),donors("Donor Is Teacher"),donors
    ↪ ("Donor Zip"))
val tmp = donor.join(donations,$"id" === $"Donor ID", "inner")
```

V proměnné tmp mám nyní uložený výsledek joinu, který použiju na agregaci. Momentálně jsou v tabulce řádky se stejným ID dárce, jelikož jeden dárce může mít několik darů. Já jsem chtěl vedět kolik každý dárce daroval dohromady, nikoliv jednotlivé dary, takže jsem tabulku agregoval dle potřeby následujícím příkazem.

```
val grouped = tmp.groupBy("id","Donor City","Donor State","Donor Is
    ↪ Teacher","Donor Zip").agg(sum("Donation Amount"))
```

S výsledkem budu dále pracovat ve vizualizačním nástroji Ksibana, tudíž si ho uložím na svůj hostovský PC. Toho dosáhnu nejprve přesunutím na HDFS a pak z HDFS do složky v dockerovského image která je namountovaná na složku v hostovském PC. Přesun do HDFS ze Sparku se provádí následovně.

```
grouped.repartition(1).write.format("com.databricks.spark.csv").option
    ↪ ("header", "true").save("hdfs://172.17.0.5:9000/test/
    ↪ DonorsDonations.csv")
```

3.4 Agregace předchozí agregace

V příkazové řádce kde máme spustěný container s HDFS napíšeme následující příkaz.

```
hadoop fs -copyToLocal /test/DonorsDonations.csv /tmp/datasets/
    ↪ DonorsDonations
```

Posledním agregačním úkolem bylo vytvořit nový dataset, který bude agregovat data ze dvou datasetů najednou, z čehož jeden bude výsledkem předchozí agregace a uložit ho zpět do databáze/na file systém.

```
val donationByCity = grouped.groupBy("Donor City").agg(sum("sum(
    ↪ Donation Amount)"))
val res = result.select(result("Donor City").as("city"),result("count
    ↪ "))
val finalResult = res.join(donationByCity,$"city" === $"Donor City", "
    ↪ inner")
val fResult = finalResult.select(finalResult("Donor City"),finalResult
    ↪ ("count").as("Number of donors"),finalResult("sum(sum(Donation
    ↪ Amount))").as("Donation Amount"))
fResult.repartition(1).write.format("com.databricks.spark.csv").option
    ↪ ("header", "true").save("hdfs://172.17.0.5:9000/test/
    ↪ DonationsByCity.csv")
```

Opět v příkazové řádce s HDFS napíšeme nasledující kopírující příkaz

```
hadoop fs -copyToLocal /test/DonationsByCity.csv /tmp/datasets/
    ↪ DonationsByCity
```

Nakonec provedeme následující kontrolní příkaz, který výsledek uložené agregace nahraje zpět do Sparku.

```
val temporary = spark.sqlContext.read.format("csv").option("header", "
    ↪ true").option("inferSchema", "true").load("hdfs
    ↪ ://172.17.0.5:9000/test/DonationsByCity.csv")
```

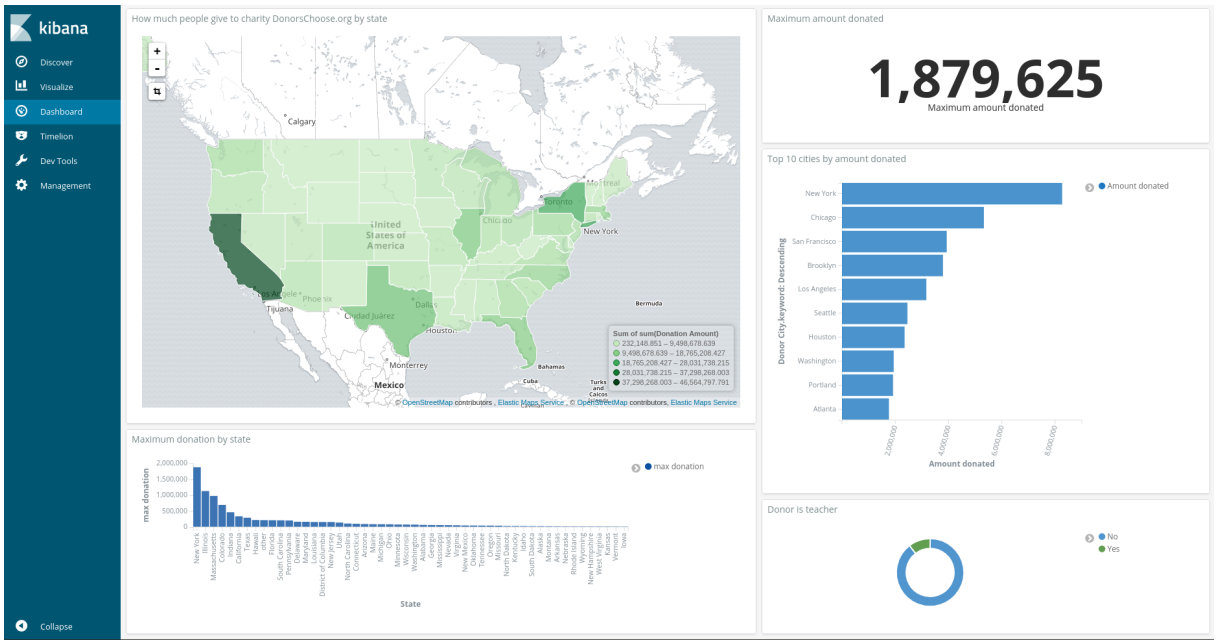
3.5 Index v Elasticsearch

Pro tvorbu indexu v Elasticsearch jsem si zvolil dataset DonorsDonations o jehož vytváření si můžete přečíst v podsekci "Agregace na dvou datasetech". Po vytvoření příslušných Logstashových konfiguračních souborů (především Logstash pipeline, která obsahuje který csv soubor a jakým způsobem nahrát a transformovat do ElasticSearch indexu) stačilo pouze spustit docker-compose.

```
cd BIG-FinalProject
docker-compose up -d
```

3.6 Vizualizace dat

Na vizualizaci dat z datasetu DonorsDonations jsem použil nástroj Kibana. V Kibaně jsem si v sekci "Management"zvolil příslušný index a poté vytvořil několik vizualizací v sekci "Visualization". Na závěr jsem tyto vizualizace naskládal do jedné obrazovky v sekci "Dashboard".



Mapa vlevo nahoře ukazuje součet darů všech lidí podle státu. Číslo vpravo nahoře ukazuje maximální součet darů od jednoho člověka. Pie chart vpravo dole ukazuje procentuální zastoupení učitelů mezi dárci a zbylé dva grafy porovnávají jednotlivá města podle součtu všech darů nebo maximálního daru. Obrázek v detailnější podobě najdete na adrese
https://www.dropbox.com/s/2p2a4e0fatuc5xu/dashboard.pdf?dl=0.

3.7 Dotazy nad indexem

Příkazy jsem zadával do Dev Tools v Kibaně.

3.7.1 Filtrování

Vypiš všechny dárce z města Hunstville.

```
GET donationsbydonors/_search
{
  "query": {
    "match": {
      "Donor City": "Huntsville"
    }
  }
}
```

3.7.2 Třídění

Seřaď všechny dárce z města Hunstville podle velikosti součtu jejich darů.

```
GET donationsbydonors/_search
{
  "sort":{
    "sum(Donation Amount)": { "order":"desc" }
    },
  "query": {
    "match": {
      "Donor City": "Huntsville"
    }
  }
}
```

3.7.3 Wildcard hledání

Najdi všechny dárce kteří pochazejí z města končící na "ville".

```
GET donationsbydonors/_search
{
  "query": {
    "wildcard": {
      "Donor City": "*ville"
    }
  }
}
```

4 Závěr

Během tvorby semestrální práce jsem si vyzkoušel práci s docker-compose a dalšími technologiemi pro práci s "big-data". Nejvíce časově náročné bylo nahrávání dat do nástrojů (Spark, Elasticsearch), ale většina následných operací byla rychlá. Z vizualizací dat jsme zjistili, že lidé ze státu California dávájí nejvíce na charitu ze všech států USA. Také jsme zjistili, že existuje zhruba desítka státu (např. Vermont), kde nejvyšší součet darovaných částek jednoho člověka se pohyboval v řádech několika stovek, což může indikovat, že v těchto státech není velké zastoupení bohatých nebo štědrých lidí.

Rejstřík