

Hadoop for Big data

Készítette: Madarász Zsolt

Szak: PTI

Dátum: 2019.05.31.

1 Tartalom

| | | |
|---------|--|---|
| 2 | Bevezetés | 2 |
| 3 | Mi az a Hadoop? | 2 |
| 3.1 | A Hadoop fő részei: | 3 |
| 3.2 | A Hive | 3 |
| 4 | Telepítés | 3 |
| 4.1 | A szükséges szoftverek letöltése | 3 |
| 4.2 | Környezet telepítése (https://hortonworks.com) | 3 |
| 5 | A project | 4 |
| 5.1 | Leírás | 4 |
| 5.2 | A megvalósítás lépései | 4 |
| 6 | Források | 8 |
| | | |
| 1. ábra | A Hive elérése SSH-n keresztül | 4 |
| 2. ábra | . csatlakozás PUTTY-al a Hive-hoz | 4 |
| 3. ábra | Mozilista letöltése | 5 |
| 4. ábra | Mozilista kitömörítése | 5 |
| 5. ábra | az item és user elemek felmásolása | 6 |
| 6. ábra | az item és user elemek felmásolása | 6 |
| 7. ábra | A tábla kilistázása | 6 |
| 8. ábra | Adatok feltöltése a Movies táblába | 7 |
| 9. ábra | Szűrés Map reduce használatával | 7 |

2 Bevezetés

Azért választottam ezt a témát, mert érdekelnek az adatbázisos rendszerek és a nagy adathalmazok kezelése, mint megoldandó probléma. A Hadoop-ról még soha nem hallottam előtte, így kíváncsian indultam neki a feladatnak. Sose rossz, ha tanulhatok valami újat. A feladat Hadoop segítségével feldolgozni nagy mennyiségű adathalmazt, majd kinyerni belőle számunkra fontos információkat. Az információk kinyeréséhez Hive-ot használtam.

3 Mi az a Hadoop?

Az **Apache Hadoop** (Wikipedia, 2017) egy nyílt forráskódú keretrendszer, amely adat intenzív elosztott alkalmazásokat támogat. Nagy mennyiségű alacsony költségű, általánosan elérhető hardverből épített szerverfürtök építését teszi lehetővé. A Hadoop a Google MapReduce és a Google File System leírásaiból készült. A Hadoop projektet Doug Cutting és Mike Cafarella hozta létre 2005-ben. Cutting, aki akkor a Yahoo!-nál dolgozott, fiának játék elefántjáról nevezte el a projektet.

3.1 A Hadoop fő részei:

- **Hadoop Common**
 - amely a fájlrendszerrel és a operációs rendszerrel kapcsolatos absztrakciókat tartalmazza, valamint azokat a szkripteket és programokat, amelyek a Hadoop rendszer indításához szükségesek
- **MapReduce rendszer**
 - A **MapReduce** egy programozási modell nagy adathalmazok feldolgozására párhuzamosan és egy szerverfürtön elosztottan.
 - A MapReduce tartalmaz egy **map** funkciót, amely szűrést és rendezést végez, valamint egy **reduce** funkciót, amely összegzi az eredményt. A MapReduce rendszer osztja el a feladatokat a szervereken párhuzamosan futtatva azokat, irányítva minden adatátvitelt, egyúttal hibatűrést is biztosít redundancián keresztül.
 - A modellt a funkcionális programozásból ismert map és reduce funkciók inspirálták, bár a használatuk nem egészen ugyanaz a MapReduce rendszerben, mint az eredeti formában.
- **Hadoop Distributed File System** (Hadoop elosztott fájlrendszer)
 - A HDFS (Hadoop Distributed File System - Hadoop Elosztott Fájlrendszer) egy elosztott, skálázható és hordozható fájlrendszer, amelyet a Hadoop rendszerhez írtak Java nyelven.

3.2 A Hive

A Hive (IBM, 2014) egy keretrendszer, amit a Hadoop köré terveztek megkönnyítve ezzel a munkánkat. A Hive-ban lehetőségünk nyílik az adatokat adatbázisokban tárolni. Ezeket SQL utasításokkal kezelhetjük. A Hadoop adminisztrációt pedig megteszi helyettünk a Hive. Olyan mintha egy fordító lenne a géphez.

4 Telepítés

4.1 A szükséges szoftverek letöltése

A rendszerkörnyezet felállításához szükség van három szoftverre. A virtualboxra, ami emulálja a hadoop környezetet és magára a hadoop-ra a HIVE-al és a többi hadoop-os kiegészítővel. Ezen környezet eléréséhez Putty-ot használunk

Hadoop Hive-al telepítő link:

<https://www.cloudera.com/downloads/ Hortonworks-sandbox/hdp.html>

Virtualbox telepítő link:

<https://www.virtualbox.org/wiki/Downloads>

Putty telepítő link:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

4.2 Környezet telepítése

1. Virtualbox feltelepítése (<https://hortonworks.com>)
2. Hortonworks implementálása: **File -> Import Appliance**, itt image file kiválasztása és **open**.
3. **Hortonworks** elindítása a **start**-al
4. A bootolás után be kell állítani a memóriaméretet, ami 22GB **System -> Motherboard -> Base Memory -> OK**

5. A Hadoop használatához SSH-n keresztül kell csatlakozni, ehhez először telepíteni kell a Connected Data Architecture (CDA)-t. Ehhez a következő sort kell begépelni: **ssh root@sandbox-hdp.hortonworks.com -p 2200**
6. Script futtatása: **cd /sandbox/deploy-scripts/**
7. majd a script futtatása: **sh enable -vm -cda.sh**

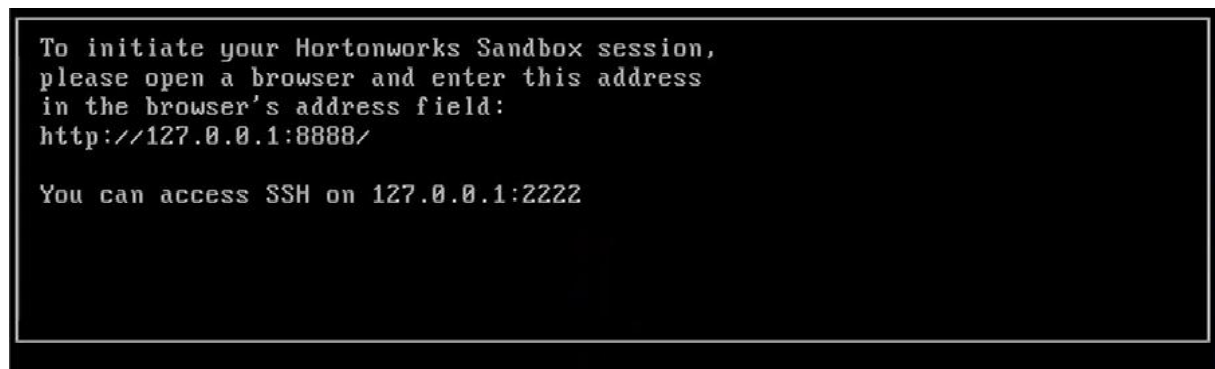
5 A project

5.1 Leírás

A feladat Hadoop alkalmazásával nagy adatmennyiség feldolgozása. Egy, vagy több nagyméretű állomány letöltése és a file tartalmának feldolgozása a Hadoop architektúra szerint. Jelen esetben a Hive rendszer használatával. A Hive adatbázisba gyűjti az adatokat és azután egyszerű SQL hívásokkal lehet kinyerni a kívánt információt.

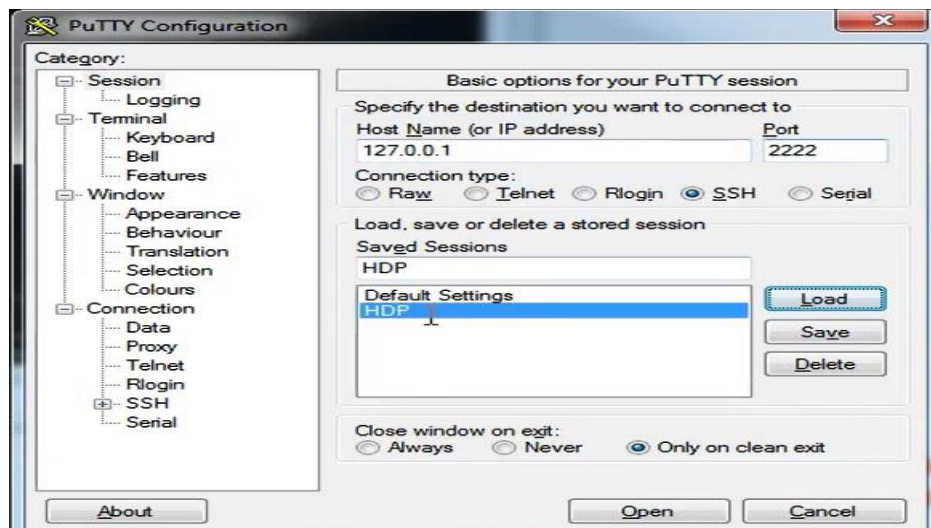
5.2 A megvalósítás lépései

1. Miután a Virtualbox befejezte a Hadoop rendszer telepítését az **1. ábra** mutatja, mit kell látnunk. Ha hasonló képet kapunk, akkor sikeres volt a telepítés. A csatlakozáshoz itt található az IP cím és a port.



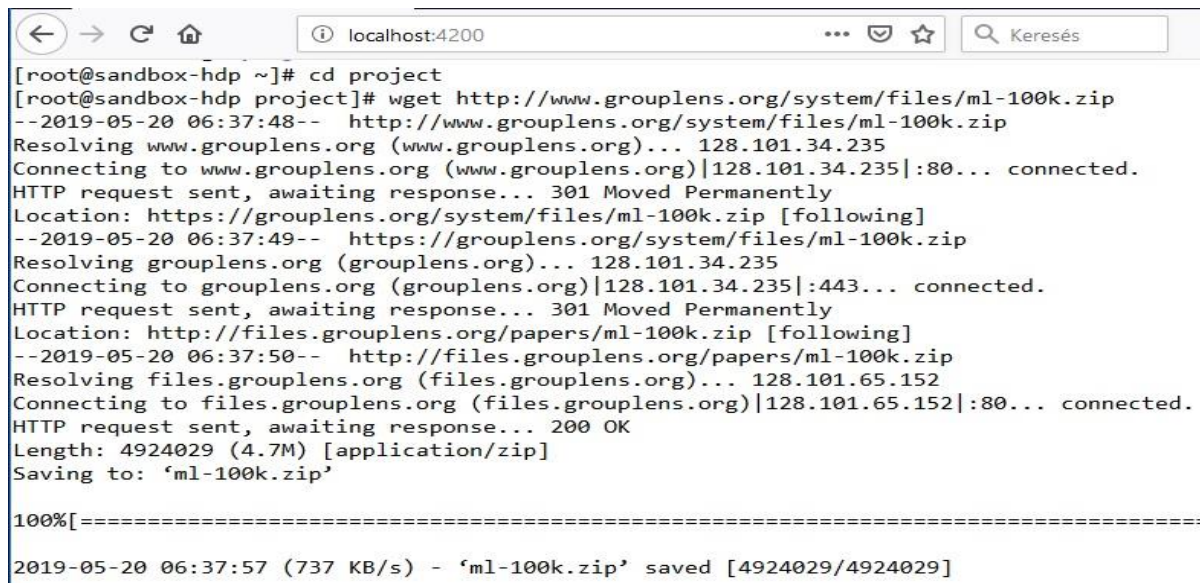
1. ábra A Hive elérése SSH-n keresztül

2. Miután kinéztük az IP címet nyissuk meg a Putty programot. Kitölteni értelemszerűen kell. Az IP címhez a címet írjuk a porthoz pedig a portot. Ezt mutatja a **2. ábra**.



2. ábra . csatlakozás PUTTY-al a Hive-hoz

3. A Hadoop Linux alapú, így linuxos parancsokat kell használni. Én létrehoztam egyproject könyvtárat. Ebbe letöltöttem a projekthez használatos fájlokat. Mivel mozilistát elemzünk, így egy hivatalos mozilistát töltöttem le a **WGET** parancssal. Az eredmény és a pontos parancsot a **3. ábra** szemlélteti.



```
[root@sandbox-hdp ~]# cd project
[root@sandbox-hdp project]# wget http://www.grouplens.org/system/files/ml-100k.zip
--2019-05-20 06:37:48-- http://www.grouplens.org/system/files/ml-100k.zip
Resolving www.grouplens.org (www.grouplens.org)... 128.101.34.235
Connecting to www.grouplens.org (www.grouplens.org)|128.101.34.235|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://grouplens.org/system/files/ml-100k.zip [following]
--2019-05-20 06:37:49-- https://grouplens.org/system/files/ml-100k.zip
Resolving grouplens.org (grouplens.org)... 128.101.34.235
Connecting to grouplens.org (grouplens.org)|128.101.34.235|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://files.grouplens.org/papers/ml-100k.zip [following]
--2019-05-20 06:37:50-- http://files.grouplens.org/papers/ml-100k.zip
Resolving files.grouplens.org (files.grouplens.org)... 128.101.65.152
Connecting to files.grouplens.org (files.grouplens.org)|128.101.65.152|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4924029 (4.7M) [application/zip]
Saving to: 'ml-100k.zip'

100%[=====
2019-05-20 06:37:57 (737 KB/s) - 'ml-100k.zip' saved [4924029/4924029]
```

3. ábra Mozilista letöltése

4. Letöltés után a file tartalmát ki kell tömöríteni. Ezt szimplán az UNZIP parancssal megtehetjük. A **4. ábra** mutatja a kitömörített fájlokat. Nekünk csk két file kell. Az item tartalmazza a mozifilmeket kategóriák szerint, az user pedig felhasználók adatait. Ezt a két filet fel kell másolni a Hadoop filerendszerbe. Ez a PUT parancssal elvégezhető. Ezt szemlélteti az **5. ábra**.

```
[root@sandbox-hdp project]# unzip ml-100k.zip
Archive: ml-100k.zip
  creating: ml-100k/
  inflating: ml-100k/allbut.pl
  inflating: ml-100k/mku.sh
  inflating: ml-100k/README
  inflating: ml-100k/u.data
  inflating: ml-100k/u.genre
  inflating: ml-100k/u.info
  inflating: ml-100k/u.item
  inflating: ml-100k/u.occupation
  inflating: ml-100k/u.user
  inflating: ml-100k/u1.base
  inflating: ml-100k/u1.test
  inflating: ml-100k/u2.base
  inflating: ml-100k/u2.test
  inflating: ml-100k/u3.base
  inflating: ml-100k/u3.test
  inflating: ml-100k/u4.base
  inflating: ml-100k/u4.test
  inflating: ml-100k/u5.base
  inflating: ml-100k/u5.test
  inflating: ml-100k/ua.base
  inflating: ml-100k/ua.test
  inflating: ml-100k/ub.base
  inflating: ml-100k/ub.test
[root@sandbox-hdp project]#
```

4. ábra Mozilista kitömörítése


```
[root@sandbox-hdp ~]# cd /home/project/
[root@sandbox-hdp project]# ls
ml-100k  ml-100k.zip
[root@sandbox-hdp project]# cd ml-100k
[root@sandbox-hdp ml-100k]# hadoop fs -put u.item project/movies
[root@sandbox-hdp ml-100k]# hadoop fs -put u.user project/userinfo
```

5. ábra az item és user elemek felmásolása

5. Ha kész vagyunk a másolással ideje elindítani a HIVE-ot. Ezt egyszerűen a **hive** begépelésével tehetjük meg. Innentől kezdve SQL utasításokat is kiadhatunk. Ha Hadoop parancsokat szeretnénk futtatni, a parancsok elé ki kell írunk a **!**. Innen tudja a fordító, hogy nem Hive-os SQL parancsok fognak futni. Létre kell hoznunk egy adatbázist a **CREATE DATABASE** adatbázisnév paranccsal. Ezt követően ki kell választani az adatbázist. Ezt az **USE** paranccsal tehetjük meg. **USE** adatbázisnév. Most már létrehozhatjuk a táblákat. Az adatbázis kiválasztását és a **Movies** tábla létrehozását a **6. ábra** szemlélteti. A tábla SQL parancsai megtalálhatóak a **GIT** mellékleten. A létrehozott táblát a **SHOW TABLES** paranccsal tudjuk lekérdezni. A kilistázást szemlélteti a **7. ábra**

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> use project;
INFO : Compiling command(queryId=hive_20190520075851_61832de7-21e8-4bb0-8d2e-499ffb3b55f8): use project
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20190520075851_61832de7-21e8-4bb0-8d2e-499ffb3b55f8); Time taken: 0.034 seconds
INFO : Executing command(queryId=hive_20190520075851_61832de7-21e8-4bb0-8d2e-499ffb3b55f8): use project
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20190520075851_61832de7-21e8-4bb0-8d2e-499ffb3b55f8); Time taken: 0.009 seconds
INFO : OK
No rows affected (0.057 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> CREATE TABLE movies ( movie_id INT, movie_title STRING, release_date STRING, video_release_date STRING, imdb_url STRING, unknown INT, action INT, adventure INT, animation INT, children INT, comedy INT, crime INT, documentary INT, drama INT, fantasy INT, film_noir INT, horror INT, musical INT, mystery INT, romance INT, sci_fi INT, thriller INT, war INT, Western INT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' STORED AS TEXTFILE;
INFO : Compiling command(queryId=hive_20190520075907_b71aa9e7-1847-427b-b30f-72498caaabfa): CREATE TABLE movies ( movie_id INT, movie_title STRING, release_date STRING, video_release_date STRING, imdb_url STRING, unknown INT, action INT, adventure INT, animation INT, children INT, comedy INT, crime INT, documentary INT, drama INT, fantasy INT, film_noir INT, horror INT, musical INT, mystery INT, romance INT, sci_fi INT, thriller INT, war INT, Western INT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' STORED AS TEXTFILE
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO : Completed compiling command(queryId=hive_20190520075907_b71aa9e7-1847-427b-b30f-72498caaabfa); Time taken: 0.188 seconds
INFO : Executing command(queryId=hive_20190520075907_b71aa9e7-1847-427b-b30f-72498caaabfa): CREATE TABLE movies ( movie_id INT, movie_title STRING, release_date STRING, video_release_date STRING, imdb_url STRING, unknown INT, action INT, adventure INT, animation INT, children INT, comedy INT, crime INT, documentary INT, drama INT, fantasy INT, film_noir INT, horror INT, musical INT, mystery INT, romance INT, sci_fi INT, thriller INT, war INT, Western INT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' STORED AS TEXTFILE
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20190520075907_b71aa9e7-1847-427b-b30f-72498caaabfa); Time taken: 0.809 seconds
INFO : OK
No rows affected (1.22 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

6. ábra az item és user elemek felmásolása

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> show tables;
INFO : Compiling command(queryId=hive_20190520080100_ef4e0311-818a-49b4-ba09-a28ed0b8aea3): show tables
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=hive_20190520080100_ef4e0311-818a-49b4-ba09-a28ed0b8aea3); Time taken: 0.042 seconds
INFO : Executing command(queryId=hive_20190520080100_ef4e0311-818a-49b4-ba09-a28ed0b8aea3): show tables
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=hive_20190520080100_ef4e0311-818a-49b4-ba09-a28ed0b8aea3); Time taken: 0.014 seconds
INFO : OK
+-----+
| tab_name |
+-----+
| movies |
+-----+
1 row selected (0.074 seconds)
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2>
```

7. ábra A tábla kilistázása

6. A táblázat létrehozása után nincs más dolgunk, mint az item file tartalmát belerakni az adatbázis **Movies** táblájába. Az item a **Movies** mappába lett belerakva a **PUT** paranccsal. Ezt a **LOAD DATA INPATH '/project/movies' INTO TABLE MOVIES** paranccsal tehetjük meg. Ha jól csináltuk a táblánk megtelik adattal. Ezt szemlélteti a **8. ábra**

```

hive> LOAD DATA INPATH '/project/movies' OVERWRITE INTO TABLE movies;
Loading data to table project.movies
Moved to trash: hdfs://sandbox:8020/apps/hive/warehouse/project.db/movies
Table project.movies stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 236344, raw_data_size: 0]
OK
Time taken: 0.889 seconds
hive> SELECT * FROM movies limit 10;
OK
1      Toy Story (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)      0
1      0      0      0      0      0      0      0
2      GoldenEye (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?GoldenEye%20(1995)      0
0      0      0      0      0      0      1      0
3      Four Rooms (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995)      0
0      0      0      0      0      0      1      0
4      Get Shorty (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)      0
0      1      0      0      0      0      0      0
5      Copycat (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?Copycat%20(1995)      0
0      1      0      0      0      0      1      0
6      Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)      01-Jan-1995      1      http://us.imdb.com/Title?Yao+a+
ipo+qiao+(1995)      0      0      0      0      0      0      0
7      Twelve Monkeys (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?Twelve%20Monkeys%20(1995)
0      0      0      0      0      1      0      0
8      Babe (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?Babe%20(1995)      0
0      1      0      0      0      0      0      0
9      Dead Man Walking (1995)      01-Jan-1995      0      http://us.imdb.com/M/title-exact?Dead%20Man%20Walking%20(1995)
0      0      0      0      0      0      0      0
10     Richard III (1995)      22-Jan-1996      0      http://us.imdb.com/M/title-exact?Richard%20III%20(1995)      0
0      0      1      0      0      0      1      0
Time taken: 0.256 seconds, Fetched: 10 row(s)
hive>

```

8. ábra Adatok feltöltése a Movies táblába

- Ha ezzel kész vagyunk hozzuk létre a táblát az userinfonak is. A kód megtalálható az SQL fájlban. Töltsük be az adatokat a LOAD DATA paranccsal az userinfóba. Semmi más dolgunk nincs, mint elkezdni az adatszűrést a SELECT parancsokkal. Egyszerű lekérdezés esetén a Hive nem alkalmazza a Map reduce eljárást, de egy összetett lekérdezés esetén már igen. Ezt szemlélteti a **9. ábra**. Itt jól látható, hogy mivel ez egy összetettebb listázás a Hive utasítja a Hadoop-ot a mapreduce bekapcsolására. Ennek folyamata jól nyomonkísérhető a pirossal bekeretezett területen.

```

hive> SELECT count(*) FROM users WHERE occupation = 'artist';
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201307141745_0004, Tracking URL = http://sandbox:50030/jobdetails.jsp?jobi
Kill Command = /usr/lib/hadoop/libexec/./bin/hadoop job -kill job_201307141745_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2019-04-28 02:57:01,902 Stage-1 map = 0%, reduce = 0%
2019-04-28 02:57:06,851 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:07,864 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:08,876 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:09,887 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:10,900 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:11,911 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:12,922 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:13,934 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:15,209 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.28 sec
2019-04-28 02:57:16,696 Stage-1 map = 100%, reduce = 33%, Cumulative CPU 2.28 sec
2019-04-28 02:57:17,706 Stage-1 map = 100%, reduce = 33%, Cumulative CPU 2.28 sec
2019-04-28 02:57:18,897 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.88 sec
2019-04-28 02:57:19,913 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.88 sec
2019-04-28 02:57:20,931 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.88 sec
2019-04-28 02:57:21,948 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.88 sec
2019-04-28 02:57:22,958 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.88 sec
2019-04-28 02:57:23,973 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.88 sec
2019-04-28 02:57:24,998 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.88 sec
MapReduce Total cumulative CPU time: 5 seconds 880 msec
Ended Job = job_201307141745_0004
MapReduce Jobs Launched:
Job 0: Map: 1 Reduce: 1 Cumulative CPU: 5.88 sec HDFS Read: 22829 HDFS Write: 3 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 880 msec

```

9. ábra Szűrés Map reduce használatával

6 Források

7 Hivatkozások

Github. (2019. 05 31). *Github*. Forrás: Github: <https://github.com/zsoltix83/Hadoop.git>

<https://hortonworks.com>. (dátum nélk.). Forrás: <https://hortonworks.com/tutorial/sandbox-deployment-and-install-guide/section/1/>.

IBM. (2014). *Apache.org*. Forrás: Hive apache: <https://hive.apache.org/>

Pluralsight. (dátum nélk.). *Pluralsight*. Forrás: Pluralsight: www.pluralsight.com

Stackoverflow. (2019). *Stackoverflow*. Forrás: Stackoverflow: <https://stackoverflow.com/>

Wikipedia. (2017. augusztus 27). *wikipedia*. Forrás: https://hu.wikipedia.org/wiki/Apache_Hadoop