

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1337

Obrada podataka tehnologijom Apache Spark

Martin Matak

Zagreb, svibanj 2016.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik. Na ovoj stranici se

nalazi izvornik.

Zahvala - TODO ...:)

SADRŽAJ

1. Uvod	1
1.1. Motivacija	1
1.2. Instalacija i izvorni kodovi	2
1.2.1. Instalacija	2
1.2.2. Izvorni kodovi	3
1.3. Osnovni gradivni elementi	5
1.4. Kratki pregled	5
2. Prvi programi	7
2.1. Postavljanje temelja	7
2.1.1. Osnovni elementi aplikacije	7
2.1.2. Programiranje u Javi	7
2.2. Programiranje s RDD-ovima	8
3. Zaključak	9
Dodatak A. Postavljanje datoteke pom.xml	10

1. Uvod

1.1. Motivacija

Pametni mobilni uređaji postaju neizostavan dodatak svakog modernog čovjeka.

Većina pametnih mobilnih uređaja u sebi sadrži sljedeće senzore:

akcelerometar - elektromehanička komponenta koja mjeri sile ubrzanja;

barometar - mehanički senzor za mjerenje atmosferskog pritiska (na trenutnoj lokaciji uređaja);

senzor svjetlosti - mjeri intenzitet, tj. jačinu svjetlosti i uglavnom se nalazi s prednje strane uređaja, iznad ekrana;

senzor blizine - u stanju prepoznati situacije kada mu neki objekt stoji u blizini - ovo omogućava automatske pozive prilikom primicanja telefona licu uz zaključavanje telefona da bi onemogućili slučajno prekidanje poziva uhom ili slično;

senzor gestikulacije - prepoznaje kretnje ruke tako što detektira infracrvene zrake koje se reflektiraju - omogućuje nam djelomično upravljanje telefonom bez doticanja ekrana;

žiroskop - uređaj koji se koristi za navigaciju i merenje kutne brzine;

geomagnetski senzor - mjeri okolno geomagnetsko polje za sve tri fizičke osi i u biti služi kao kompas na mobilnim uređajima i

Hall Sensor - magnetski senzor zadužen za prepoznavanje je li maska telefona zatvorena ili otvorena.

Pretpostavimo da je mobilni uređaj spojen na internet i da svakih nekoliko sekundi pošalje vrijednost koju u tom trenutku mjeri pojedini senzor. U samo jednom danu može se skupiti dosta podataka. A što kada to ne bi radili za jedan uređaj nego za sve izdane uređaje nekog modela? Količina podataka bi jako brzo narasla.

Kako količina podataka postaje sve veća, dolazimo do pojma *Velika količina podataka* (engl. *Big Data*). U današnje vrijeme imamo više podataka u digitalnom obliku nego što smo ikada imali. Jedan od zanimljivijih izazova je kako ih efektivno obraditi i zaključiti nešto iz toga. Kako od te velike količine podataka doći do nekih pametnih zaključaka iz kojih ćemo nešto novo naučiti.

Apache Spark je otvorena (engl. *open source*) tehnologija koja omogućava pisanje programa za obradu podataka u tri programskih jezika: *Java*, *Python* i *Scala*; a nudi i mogućnost interaktivnog rada.

U okviru ovog rada biti će proučene mogućnosti ove tehnologije, razrađeno nekoliko konkretnih primjera obrade podataka te ostvarena programska rješenja koja obavljaju tu obradu koristeći *Apache Spark*.

Svi primjeri će biti napisani u programskom jeziku *Java*.

1.2. Instalacija i izvorni kodovi

1.2.1. Instalacija

Apache Spark je pisan u programskom jeziku *Scala* i izvršava se na *Javinom virtualnom stroju* (engl. *Java Virtual Machine*) (JVM). Instalacija na osobno računalo je prilično jednostavna, a u ovdje će biti prikazana instalacija na operacijskom sustavu *Ubuntu 15.10*.

Za početak je potrebno imati instaliranu Javu, a je li Java instalirana na računalo se može provjeriti tako što se u naredbenom retku unese sljedeća naredba:

`java -version`. Kao rezultat bi trebali dobiti trenutno instaliranu verziju Jave. Ukoliko Java nije instalirana, potrebno ju je najprije instalirati. Taj dio neće biti objašnjen ovdje.

Jednom kada imamo instaliranu Javu, sve što treba napraviti je otići na službene stranice: <https://spark.apache.org/downloads.html>, odabrati najnoviju verziju (Za vrijeme pisanja ovog rada to je verzija *1.6.1 (Mar 09 2016)*, izabrati odgovarajući paket te pokrenuti dohvaćanje odgovarajuće *.tgz* arhive. Najjednostavnije je odabrati neki *pre-built* paket, primjerice *Pre-built for Hadoop 2.6 and later* te će daljnji koraci instalacije biti napisani pod pretpostavkom da je korisnik dohvatio tu verziju paketa. Ukoliko korisnik želi, moguće je instalirati i *Source code* varijantu paketa, ali taj postupak instalacije ovdje nije opisan.

Nakon što smo dohvatili odgovarajuću arhivu, potrebno ju je raspakirati. Raspakiranje arhive moguće je napraviti preko naredbe:

```
$ tar -xvf spark-1.6.1-bin-hadoop2.6.tgz
```

Nakon toga, dobra je praksa premjestiti instalaciju u neki prikladniji direktorij. Tako nešto može se napraviti na sljedeći način:

```
$ mv Downloads/spark-1.6.1-bin-hadoop2.6 faks/spark/
```

Službeno, sada je instalacija gotova. Idemo u sljedećem odlomku malo detaljnije pogledati što smo to zapravo dobili instalacijom. Koje datoteke se sada nalaze na našem računalu, a nije ih bilo ranije. Također, biti će objašnjena i uloga nekih datoteka i direktorija.

1.2.2. Izvorni kodovi

Ako izlistamo što nam se trenutno nalazi u novonastalom direktoriju, dobiti ćemo sljedeći ispis:

```
mmatak@martins-beast:~/faks/spark$ ls -l
total 1408
drwxr-xr-x  2 mmatak mmatak    4096 Feb 27 06:02 bin
-rw-r--r--  1 mmatak mmatak 1343562 Feb 27 06:02 CHANGES.txt
drwxr-xr-x  2 mmatak mmatak    4096 Feb 27 06:02 conf
drwxr-xr-x  3 mmatak mmatak    4096 Feb 27 06:02 data
drwxr-xr-x  3 mmatak mmatak    4096 Feb 27 06:02 ec2
drwxr-xr-x  3 mmatak mmatak    4096 Feb 27 06:02 examples
drwxr-xr-x  2 mmatak mmatak    4096 Feb 27 06:02 lib
-rw-r--r--  1 mmatak mmatak 17352 Feb 27 06:02 LICENSE
drwxr-xr-x  2 mmatak mmatak    4096 Feb 27 06:02 licenses
-rw-r--r--  1 mmatak mmatak 23529 Feb 27 06:02 NOTICE
drwxr-xr-x  6 mmatak mmatak    4096 Feb 27 06:02 python
drwxr-xr-x  3 mmatak mmatak    4096 Feb 27 06:02 R
-rw-r--r--  1 mmatak mmatak   3359 Feb 27 06:02 README.md
-rw-r--r--  1 mmatak mmatak    120 Feb 27 06:02 RELEASE
drwxr-xr-x  2 mmatak mmatak    4096 Feb 27 06:02 sbin
```

README.md Sadrži kratke instrukcije za upoznavanje sa Spark-om.

bin Sadrži izvršive datoteke koje se koriste za interaktivni rad sa Spark-om.

Zanimljivo je spomenuti da postoji interaktivna ljuska *Spark shell* za programske jezike *Python* i *Scala*. Datoteke u ovom direktorij služe upravo za to. Budući

da je ovaj rad ograničen isključivo na programski jezik *Java*, ovaj dio neće biti detaljnije obrađen.

core, streaming, python, ... Sadrži glavne komponente projekta *Apache Spark*

examples Sastoji se od nekoliko jednostavnih primjera koje pomažu korisniku da se uhoda i što bezbolnije nauči koristiti odgovarajući API.

Kako bi bili sigurni da je instalacija uspješna, potrebno je pozicionirati se u *bin* direktorij te u terminalu upisati `./spark-shell`. Ispis bi trebao biti sličan ovome:

```
mmatak@martins-beast:~/faks/spark/bin$ ./spark-shell
```

```
Welcome to
```

```

  ____
 /  __/  __  ____  ____/  /  __
_ \  \ /  _  \ /  _ ' /  __/  ' _/
/  __/  . __/ \_, _/ _/  / _/ \_, _\   version 1.6.1
/_/

```

```
Using Scala version 2.10.5 (OpenJDK 64-Bit Server VM, Java 1.8.0_91)
```

```
Type in expressions to have them evaluated.
```

```
Type :help for more information.
```

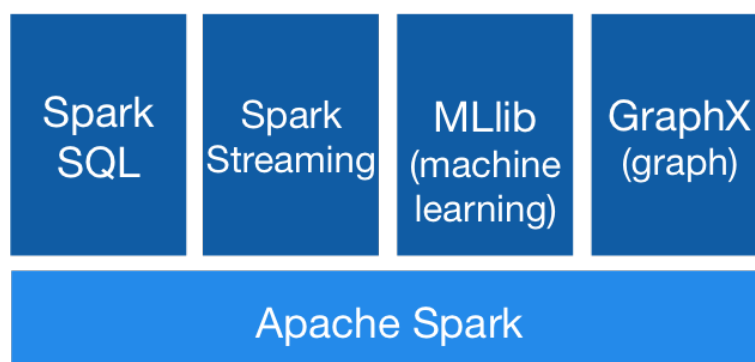
```
scala>
```

Ukoliko se prikaže greška vezana uz *sqlContext*, to je OK. U ovom trenutku to nije važno. Važno je napomenuti da su u gornjem ispisu (radi ljepšeg formata ovog rada) izbrisana upozorenja (engl. *warnings*) koja su normalna i očekivana.

1.3. Osnovni gradivni elementi

Osnovna programska apstrakcija s kojom *Apache Spark* radi naziva se *resilient distributed dataset* (RDD). RDD-ovi predstavljaju kolekcije podataka koje se nalaze na računalima (jednom ili više njih) i moguće ih je paralelno obrađivati. *Apache Spark* nudi bogati API za rad s RDD-ovima.

Apache Spark se sastoji od nekoliko ključnih elemenata koji će ovdje biti samo nabrojani i opisani rečenicom ili dvije, a u kasnijim poglavljima će biti detaljnije objašnjeni. Slika koja predstavlja osnovne elemente je sljedeća:



Slika 1.1: Osnovni elementi tehnologije *Apache Spark*.

Na slici su prikazana četiri osnovna elementa koja čine osnovu tehnologije *Apache Spark*, a to su:

Spark SQL - omogućuje rad s bilo kakvim strukturiranim podacima, primjerice JSON.

Nudi mogućnost izvršavanja SQL-a nad RDD-ovima;

Spark Streaming - komponenta zadužena za rad s tokovima podataka;

MLlib - koristi se za postupak strojnog učenja i

GraphX - biblioteka za obradu grafova (npr. graf prijatelja na društvenoj mreži).

1.4. Kratki pregled

U ovom poglavlju je bila ideja da čitatelj dobije motivaciju i želju za upoznavanjem s tehnologijom *Apache Spark*. Detaljno je opisan postupak instalacije koji bi čitatelju trebao biti sasvim dovoljan za samostalnu instalaciju. Također, dan je pregled nekih najosnovnijih datoteka i direktorija koje dolaze s instalacijom. Zainteresiranog čitatelja se ohrabruje da samostalno prouči kako se koristi *Spark shell*. Na kraju, nabrojani su

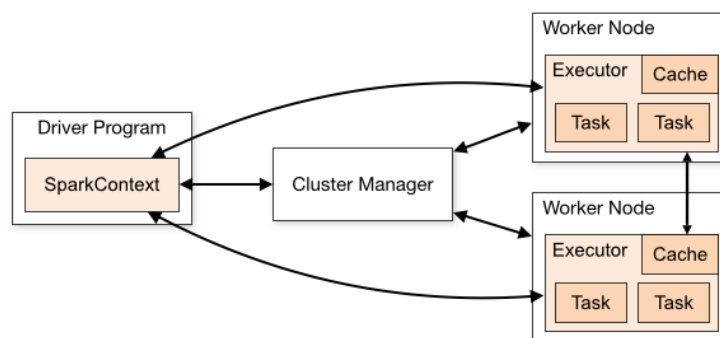
osnovni elementi koji čine jezgru tehnologije *Apache Spark* i uveden je pojam osnovne programske apstrakcije koja se koristi u ovoj tehnologiji, a to je *resilient distributed dataset* (RDD).

2. Prvi programi

2.1. Postavljanje temelja

2.1.1. Osnovni elementi aplikacije

Općenito govoreći, svaka Spark aplikacija sastoji se od nekoliko komponenata. Prva komponenta koju ćemo spomenuti je program koji se izvršava - onaj čija je `main` metoda pokrenuta, odnosno onaj koji pokreće obradu podataka. Taj program se naziva *driver*. On s podacima priča kroz "tunel" između sebe i Sparka odnosno kroz *SparkContext*. Budući da je *Apache Spark* namijenjen za paralelnu obradu podataka, idemo se odmah upoznati i s još dvije komponente. Pojedino računalo u *grozdu* (engl. *cluster*) ćemo nazvati *radnim čvorom* (engl. *worker node*), a proces koji se izvršava na pojedinom računalu nazvati ćemo *radnik* (engl. *executor*). Dozvoljeno je da *radnici* međusobno komuniciraju. U cijeloj priči može, a i ne mora eksplicitno biti uključen *upravitelj grozdom* (engl. *cluster manager*).



Slika 2.1: Prikaz elemenata aplikacije.

2.1.2. Programiranje u Javi

Kako bi mogli pisati Spark programe u programskom jeziku *Java*, potrebno je povezati svoju aplikaciju s `spark-core` artifaktom s Mavena. Za postavljanje i instalaciju

Mavena konzultirati <https://maven.apache.org/install.html> i knjigu (Programiranje u Javi, Čupić). Za vrijeme pisanja ovog rada, najnovija verzija Sparka je 1.6.1, a odgovarajuće Maven koordinate su:

```
groupId = org.apache.spark  
artifactId = spark-core_2.10  
version = 1.6.1
```

Odgovarajuća `pom.xml` datoteka nalazi se u dodatku A

2.2. Programiranje s RDD-ovima

3. Zaključak

Zaključak.

A. Postavljanje datoteke pom.xml

Datoteka koja je potrebna kako bi Maven ispravno dohvatio artifakt `spark-core` bi trebala izgledati slično kao što je dano u nastavku. Za ispravno funkcioniranje, trebala bi se zvati `pom.xml` i nalaziti se direktno u direktoriju `src`.

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>hr.fer.zemris</groupId>
  <artifactId>Prva Spark aplikacija</artifactId>
  <packaging>jar</packaging>
  <version>1</version>
  <name>Prva Spark aplikacija</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>org.apache.spark</groupId>
      <artifactId>spark-core_2.10</artifactId>
      <version>1.6.1</version>
    </dependency>
  </dependencies>
</project>
```

Obrada podataka tehnologijom Apache Spark

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.