Diplomski studij

Informacijska i komunikacijska tehnologija

Telekomunikacije i informatika

Računarstvo

Programsko inženjerstvo i informacijski sustavi

Računalno inženjerstvo
Računarska znanost

Ak. g. 2017./2018.

Raspodijeljena obrada velike količine podataka

4. Laboratorijska vježba

Zadatak 1: Rad s kolekcijskim tokovima

Zadatak

Cilj zadatka je napisati, prevesti te izvršiti Javin program koji će učitati podatke iz deset tekstualnih datoteka, filtrirati ih i sortirati te zapisati na lokalni disk. Ovaj program treba koristiti kolekcijske tokove iz *Java 8 Streams API*-ja. Ulazne tekstualne datoteke sadrže podatke o zagađenju zraka. Svaka datoteka sadrži očitanja s jedne mjerne postaje pa je cilj zadatka dobiti jednu izlaznu sortiranu datoteku s očitanjima sa svih senzorskih postaja. Ova izlazna datoteka će biti korištena u 3. zadatku kao ulaz generatora toka senzorskih podataka.

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **osnove rada s kolekcijskim tokovima iz *Java 8 Streams API*-ja**
- **jednostavna predobrada ulaznih podataka**

Detaljni opis zadatka

Za potrebe ovog zadatka dohvatite ulaznu datoteku sa sljedeće poveznice: http://svn.tel.fer.hr/pollutionData.zip. U arhivi se nalaze datoteke sa senzorskim očitanjima koje imaju naziv u obliku `pollutionDataxxxxxx.csv`, gdje je `xxxxxx` redni broj (ID) mjerne postaje.

U ovom zadatku ćete napraviti Javin program će učitati linije svih ulaznih datoteka `pollutionDataxxxxxx.csv` u jedan jedinstveni kolekcijski tok linija. Pri tome koristite metodu `list` iz klase `Files`. Nakon toga je potrebno iz ovog kolekcijskog toka izbaciti (profiltrirati) sve linije koje se ne mogu parsirati. Filtrirani tok linija je nakon toga potrebno pretvoriti u filtrirani tok očitanja (vlastita klasa `PollutionReading`) kojeg je zatim potrebno sortirati po vremenu očitanja (pri definiranju komparatora koristite metodu `comparing` iz funkcijskog sučelja `Comparator` i operator *double-colon ::*) i zapisati u jednu jedinstvenu izlaznu tekstualnu datoteku `pollutionData-all.csv`. Format ove datoteke treba biti CSV, na način da su parametri očitanja odvojeni zarezom (u redoslijedu kakav je u ulaznim datotekama), a u svakom retku je drugo senzorsko očitanje.

Zadatak 2: Obrada podataka programskim okvirom Apache Spark

Zadatak

Cilj zadatka je uspješno napisati, prevesti te izvršiti Javin program koji će obaviti analizu podatka o umrlima u Sjedinjenim Američkim Državama.

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **osnove rada s programskim okvirom Apache Spark (Core)**
- **jednostavna analiza velike količine podataka**

Detaljni opis zadatka

U ovom zadatku ćete napraviti Javin program u obliku Maven projekta u koji je potrebno uključiti Apacheov paket `spark-core` kao *dependency*. Arhivu s podacima koji su neophodni za ovaj zadatak dohvatite sa sljedeće poveznice: http://svn.tel.fer.hr/DeathRecords.csv. U arhivi se nalazi datoteka DeathRecords.csv s podacima o umrlim osobama u Sjedinjenim Državama u 2014. godini. Svaka linija datoteke predstavlja zapis u sljedećem obliku (definirano je u prvoj liniji datoteke): `Id, ResidentStatus, Education1989Revision, Education2003Revision, EducationReportingFlag, MonthOfDeath, Sex, AgeType, Age, AgeSubstitutionFlag, AgeRecode52, AgeRecode27, AgeRecode12, InfantAgeRecode22, PlaceOfDeathAndDecedentsStatus, MaritalStatus, DayOfWeekOfDeath, CurrentDataYear, InjuryAtWork, MannerOfDeath, MethodOfDisposition, Autopsy, ActivityCode, PlaceOfInjury, Icd10Code, CauseRecode358, CauseRecode113, InfantCauseRecode130, CauseRecode39, NumberOfEntityAxisConditions, NumberOfRecordAxisConditions, Race, BridgedRaceFlag, RaceImputationFlag, RaceRecode3, RaceRecode5, HispanicOrigin, HispanicOriginRaceRecode`. Polje `MonthOfDeath` predstavlja redni broj mjeseca u kojem je osoba umrla, polje `Sex` predstavlja spol u obliku `M` za mušku osobu i `F` za žensku osobu, polje `Age` predstavlja godine koje je osoba imala u trenutku smrti, polje `MaritalStatus` predstavlja bračni status u obliku `M` za udanu osobu, `D` za rastavljenu osobu i `W` za udovca/icu, polje `DayOfWeekOfDeath` predstavlja redni broj dana u tjednu, polje `MannerOfDeath` označava način na koji je osoba umrla, a polje `Autopsy` označava je li nakon smrti provedena autopsija u obliku `Y` za da, `N` za ne i `U` za nepoznato. Za parsiranje datoteke `DeathRecords.csv` koristite gotovu klasu `USDeathRecord`. Datoteku je potrebno učitati u jedan jedinstveni RDD *(Resilient Distributed Dataset*). Nakon toga je potrebno iz RDD-a izbaciti (profiltrirati) sve linije koji se ne mogu parsirati (npr. prva linija). Filtrirani RDD s linijama je nakon toga potrebno pretvoriti u filtrirani RDD sa zapisima o umrlima u Sjedinjenim Američkim Državama. U nastavku napišite programski kod koji će obraditi dobiveni RDD i (jedno po jedno) dati odgovore na sljedeća pitanja:

1. **Koliko je ženskih osoba starijih od 40 godina umrlo kroz čitav period?**
2. **Koji mjesec u godini je umrlo najviše muških osoba mlađih od 50 godina?**
3. **Koliko ženskih osoba je bilo podvrgnuto obdukciji nakon smrti?**
4. **Kakvo je kretanje broja umrlih žena u dobi između 50 i 65 godina po danima u tjednu?** Rezultat je (sortirana) lista tipa `Pair2` (ključ je redni broj dana, a vrijednost je broj umrlih žena)
5. **Kakvo je kretanje postotka umrlih udanih žena u dobi između 50 i 65 godina po danima u tjednu?** Rezultat je (sortiran) skup tipa `Pair2` (ključ je redni broj dana, a vrijednost je postotak).
6. **Koji je ukupni broj muškaraca umrlih u nesreći (kod 1) u cjelokupnom periodu**?
7. **Koliki je broj različitih godina starosti umrlih osoba koji se pojavljuju u zapisima**?

NAPOMENA: Koristite priručno spremanje RDD-ova da izbjegnete njihovo ponovno učitavanje kao što je objašnjeno na poveznici: https://spark.apache.org/docs/latest/programming-guide.html#rdd-persistence. U rješenju koristite programski okvir Apache Spark što je više moguće, a Javine kolekcije podataka samo za pohranu konačnog rezultata (ako je to zadano).

Aplikaciju pokrenite na grozdu u laboratoriju.

Zadatak 3: Obrada toka podataka programskim okvirom Apache Spark

Zadatak

Cilj zadatka je uspješno napisati, prevesti te izvršiti Javin program koji će obaviti obradu toka senzorskih podataka.

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

- **osnove rada s programskim okvirom Apache Spark (Streaming)**
- **jednostavna obrada toka podataka**

Detaljni opis zadatka

U ovom zadatku ćete napraviti Javin program u obliku Maven projekta u koji je potrebno uključiti Apacheov paket `spark-streaming` kao *dependency*. Za rješavanje zadatka koristite generator toka senzorskih podataka (`SensorStreamGenerator.java`) iz 4. domaće zadaće koji koristi ulaznu datoteku koju ste dobili u prvom zadatku. Nakon što se na njega poveže klijent (TCP tok na portu `10002`), generator proizvodi senzorska očitanja intenzitetom od 1 očitanja u milisekundi. Vaš zadatak je obraditi ovaj tok podataka u mikro-skupinama od očitanja pristiglih u 3 sekunde na način da ćete prvo iz ovog toka izbaciti (profiltrirati) sve linije koje se ne mogu parsirati. Filtrirani tok linija je nakon toga potrebno pretvoriti u filtrirani tok očitanja (vlastita klasa `PollutionReading` iz 1. zadatka) kojeg je zatim potrebno pretvoriti u tok parova kod kojega je ključ `stationID`, a vrijednost `ozone`. Ključ `stationID` je geografska lokacija postaje (latitude longitude). Nakon toga, za svaki stationID izračunajte minimalni `ozone` u prozoru veličine 45 sekundi koji se izračunava svakih 15 sekundi. Neka ove minimalne vrijednosti također budu u obliku toka parova kod kojega je ključ `stationID`, a vrijednost `ozone`. Rezultat pohranite na disk kao što je objašnjeno na predavanju.

IZVJEŠTAJ: **Na sustav Moodle potrebno je predati izvještaj koji sadrži rješenja svih zadataka (odgovore na pitanja i pripadajući izvorni programski kod) do**

ponedjeljka 18.06.2018. u 09:00 sati