Diplomski studij

Informacijska i komunikacijska tehnologija

Telekomunikacije i informatika

Računarstvo

Programsko inženjerstvo i informacijski sustavi

Računalno inženjerstvo
Računarska znanost

Ak. g. 2016./2017.

Raspodijeljena obrada velike količine podataka

4. Domaća zadaća

1. Zadatak: Rad s kolekcijskim tokovima

Cilj zadatka je uspješno napisati, prevesti te izvršiti Javin program koji će učitati podatke iz nekoliko 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 senzorska očitanja koja su prikupljena projektom Sensorscope. 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**

> **IZVJEŠTAJ:** Na sustav Moodle predajete izvorni programski kôd te odgovore na pitanja koja slijede iza opisa zadataka.

Za potrebe ove zadaće će biti potrebno dohvatiti arhivu `sensorscope-monitor.zip` s ulaznim podacima sa sljedeće poveznice: http://lcav.epfl.ch/page-86035-en.html. Opis podataka se nalazi u samoj arhivi u datoteci `sensorscope-monitor-def.txt`. Ostatak arhive su datoteke sa senzorskim očitanjima koje imaju naziv u obliku `senesorscope-monitor-xx.txt`, gdje je xx redni broj (ID) mjerne postaje.

Detaljni opis zadatka:

U ovom zadatku ćete napraviti Javin program će učitati linije svih ulaznih datoteka `senesorscope-monitor-xx.txt` u jedan jedinstveni kolekcijski tok linija. Njega ćete napraviti na način kako je to objašnjeno na poveznici https://stackoverflow.com/questions/29691209/is-there-any-way-for-reading-two-or-more-files-in-one-java8-stream. 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 `SensorscopeReading`) 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 `senesorscope-monitor-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.

Nakon uspješnog pokretanja programa odgovorite na sljedeća pitanja:

- Koliko je bilo ulaznih datoteka `senesorscope-monitor-xx.txt`?
- Koliko se zapisa nalazi u izlaznoj datoteci?
- Kolika je veličina izlazne datoteke?

2. Zadatak: Obrada podataka programskim okvirom Apache Spark

Cilj zadatka je uspješno napisati, prevesti te izvršiti Javin program koji će obaviti analizu podatka o učestalosti imena novorođenčadi u Sjedinjenim Američkim Državama po godinama i državama.

Uspješnim rješenjem ovog zadatka steći ćete sljedeća znanja:

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

IZVJEŠTAJ: Na sustav Moodle predajete izvorni programski kôd te odgovore na pitanja koja slijede iza opisa zadataka. **Pitanja koja zahtijevaju prikaz kretanja nekog parametra kroz niz godina prikažite na grafu.**

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/StateNames.csv.zip. U arhivi se nalazi datoteka StateNames.csv s učestalosti imena novorođenčadi u Sjedinjenim Američkim Državama po godinama i državama. Svaka linija datoteke predstavlja zapis u sljedećem obliku (definirano je u prvoj liniji datoteke): `Id`, `Name`, `Year`, `Gender`, `State`, `Count`. Polje `Id` predstavlja redni broj zapisa, polje `Gender` predstavlja spol u obliku `M` za muško dijete i `F` za žensko dijete, a oznaka države je u USPS obliku od dva slova, npr. CA za Kaliforniju. 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 imenima novorođenčadi (vlastita klasa `USBabyNameRecord`). U nastavku napišite programski kod koji će obraditi dobiveni RDD i (jedno po jedno) dati odgovore na sljedeća pitanja:

1. **Koje je najnepopularnije žensko ime kroz čitav period i države?**
2. **Kojih 10 muških imena su najpopularnija kroz čitav period i države?**
3. **U kojoj državi je 1946. godine rođeno najviše djece oba spola?**
4. **Kakvo je kretanje broja novorođene ženske djece kroz godine?** Rezultat je (sortirana) lista tipa `Pair2` (ključ je godina, a vrijednost je broj novorođenčadi)
5. **Kakvo je kretanje postotka imena Mary kroz godine?** Rezultat je (sortiran) skup tipa `Pair2` (ključ je godina, a vrijednost je postotak). Pri tome iskoristite polje iz prethodnog pitanja.
6. **Koji je ukupni broj rođene djece u cjelokupnom periodu u svim državama**?
7. **Koliki je broj različitih imena koja 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).

Instalirajte i podesite Spark na svom pseudo-raspodijeljenom grozdu kako je objašnjeno na predavanju. Pokušajte pokrenuti aplikaciju na njemu jer će se to tražiti na laboratorijskoj vježbi.

3. Zadatak: Obrada toka podataka programskim okvirom Apache Spark

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**

IZVJEŠTAJ: Na sustav Moodle predajete izvorni programski kôd te odgovore na pitanja koja slijede iza opisa zadataka.

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*. Zadatku je priložen generator toka senzorskih podataka (`SensorStreamGenerator.java`) 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 5 sekundi 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 `SensorscopeReading` iz 1. zadatka) kojeg je zatim potrebno pretvoriti u tok parova kod kojega je ključ `stationID`, a vrijednost `solarPanelCurrent`. Nakon toga, za svaki stationID izračunajte maksimalni `solarPanelCurrent` u prozoru veličine 60 sekundi koji se izračunava svakih 10 sekundi. Neka ove maksimalne vrijednosti također budu u obliku toka parova kod kojega je ključ `stationID`, a vrijednost `solarPanelCurrent`. Rezultat pohranite na disk kao što je objašnjeno na predavanju.

Nakon uspješnog pokretanja odgovorite na sljedeća pitanja:

- Koliko često (u sekundama) nastaje novi direktorij na disku?
- Koliko često (u sekundama) se pokreće izračun?
- Može li vrijednost parametra `solarPanelCurrent` neke stanice biti manja u nekom direktoriju nego u njegovom neposrednom prethodniku. Zašto?
- Kako se kreću vrijednosti parametra `solarPanelCurrent` neke postaje u prva 3 direktorija koji su nastali? Zašto?