Instalacija programske podrške

Prije instalacije programske podrške korištene u ovom diplomskom radu, potrebno je instalirati alate:

- Oracle Java JDK 1.8.0
- Apache Kafka 2.11
- Apache Zookeper 3.6.2
- Apache Spark 2.4.7.
- Apache Hadoop 2.7.
- Apache Flink 1.10.0
- Apache Maven 3.6.3

Opcionalno se mogu instalirati alati:

- Python 3.7
- Hazelcast Jet 4.3
- Scala 2.11

Ulaznu datoteku, koja se koristila prilikom izrade projekta, dostupna je na poveznici:

- SMART GREEN INFRASTRUCTURE MONITORING SENSORS – HISTORICAL, https://data.cityofchicago.org/api/views/ggws-77ih/rows.csv?accessType=DOWNLOAD&bom=true&format=true

Programski kod dostupan je na idućoj poveznici:

- https://github.com/mateoz-28/apache-beam-test

Potrebno je preuzeti programski kod u obliku *.zip* datoteke te ju raspakirati pomoću 7zip ili sličnih alata za raspakiravanje sadržaja. Nakon što je sadržaj raspakiran u lokalni direktorij, potrebno je konfigurirati put do ulazne *.csv* datoteke. U datoteci *Constants* potrebno je postaviti varijablu *INPUT_FILE_PATH* tako da sadrži put do *.csv* datoteke. Nakon što je varijabla postavljena na ispravnu vrijednost, potrebno je otvoriti naredbeni redak, pozicionirati se u direktorij gdje je raspakirana programska podrška te pokrenuti iduće naredbe:

```
➢ mvn compile
➢ mvn install
```

Kod 7.1 Naredbe za izgradnju projekta

Projekt bi sada trebao biti uspješno izgrađen te spreman za korištenje.

Upute za korištenje programske podrške

Napomena: upute su napisane za operacijski sustav Windows 10, stoga se uputstva mogu razlikovati ako se koristi drugi operacijski sustav za pokretanje programa.

Prije pokretanja programa, potrebno je provjeriti jesu li idući parametri ispravno konfigurirani:

- Varijabla okoline *JAVA_HOME* pokazuje na direktorij gdje je instaliran programski jezik Java
- Varijabla okoline *SPARK_HOME* pokazuje na direktorij gdje je instaliran alat Apache Spark
- Varijabla okoline *FLINK_HOME* pokazuje na direktorij gdje je instaliran alat Apache Flink
- Varijabla okoline *FLINK_CONF_DIR* pokazuje na direktorij gdje se nalaze konfiguracijske datoteke za alat Apache Flink
- Varijabla okoline *ZOOKEPER_HOME* pokazuje na direktorij gdje je instaliran alat Apache Zookeper
- Varijabla okoline *HADOOP_HOME* pokazuje na direktorij gdje je instaliran alat *winutils.exe*
- Varijabla okoline *Path* sadrži put do direktorija gdje je instaliran alat Apache Maven

Sve gore navedene varijable okoline potrebno je nadodati u *Path* varijablu okolinu ako se tamo ne nalaze.

Glavni program koristi alat Apache Kafka za čitanje i pisanje podataka, stoga je potrebno pokrenuti alate Apache Zookeper i Apache Kafka prije pokretanja glavnog programa. Za pokretanje alata, potrebno je otvoriti naredbeni redak te izvršiti iduću naredbu:

```
➢ zkserver
```

Kod 7.2 Naredbe za pokretanje alata Apache Zookeper

Nakon pokretanja Zookepera, potrebno se pozicionirati u direktorij gdje je instaliran alat Apache Kafka. U konfiguracijskoj datoteci *server.properties* potrebno je postaviti varijablu *log.dirs* tako da pokazuje na direktorij gdje će se spremati zapisnik. Nakon konfiguriranja

parametra, potrebno je pokrenuti naredbeni redak, pozicionirati se u direktorij gdje je instaliran alat te pokrenuti iduću naredbu:

```
➢ .\bin\windows\kafka-server-start.bat .\config\server.properties
```

Kod 7.3 Naredba za pokretanje alata Apache Kafka

Konfiguracijska datoteka *Constants* sadrži varijable *INPUT_TOPIC_NAME* i *OUTPUT_TOPIC_NAME* koje imenuju naziv Kafkine teme za čitanje ulaznih podataka i slanje rezultata izvršavanja glavnog programa. Ako teme nisu prethodno stvorene, potrebno je pozvati naredbe za njihovo stvaranje. U naredbenom retku potrebno je pozicionirati se u direktorij gdje je instaliran alat Apache Kafka te pokrenuti iduću naredbu:

```
➢ .\bin\windows\kafka-topics.bat --create --topic input --replication-factor 1
  --partitions 1 --bootstrap-server localhost:9092
```

Kod 7.4 Naredba za kreiranje teme alata Apache Kafka

Naredba sadrži više konfiguracijskih opcija koje se mogu prilagoditi vlastitim potrebama, ali najvažnije je da parametar *topic* odgovara nazivu teme iz konfiguracijske datoteke *Constants*.

Nakon što je alat Apache Kafka ispravno konfiguriran i pokrenut, potrebno je pozicionirati se u direktorij gdje se nalazi projekt te pokrenuti naredbeni redak. Naredba za pokretanje glavnog programa ima iduću oblik:

```
➢ mvn compile exec:java -D exec.mainClass=hr.mateo.beam.Beam -D exec.args="--r
  unner=FlinkRunner --task=aggregation" -P flink-runner
```

Kod 7.5 Primjer naredbe za pokretanje alata Apache Beam

Parametar *runner* definira koji se program pokretač koristi za obradu toka podataka. Trenutno taj parametar može poprimiti iduće vrijednosti:

- *FlinkRunner*
- *SparkRunner*
- *JetRunner*
- *Twister2Runner*
- *SamzaRunner*
- *DirectRunner*

Parametar *task* definira operaciju koja se izvršava i može poprimiti jednu od idućih vrijednosti:

- *aggregation*

- *specificValues*
- *filtration*
- *identity*
- *topN*
- *compositeTransformation*

Ako korisnik ne definira ovaj parametar, program će izvršiti operaciju identitet.

Parametar definiran iza „*-P*“ imenuje profil koji se koristi. Profil mora odgovarati programu pokretaču koji se definirao s argumentom *runner* i može poprimiti iduće vrijednosti:

- *flink-runner*
- *spark-runner*
- *jet-runner*
- *twister2-runner*
- *samza-runner*
- *direct-runner*

Naredba za pokretanje glavnog programa može imati dodatne parametre specifične za korišteni program pokretač. Detaljan popis mogućih parametra dostupan je na idućoj poveznici:

- Beam Capability Matrix, https://beam.apache.org/documentation/runners/capability-matrix/

Na poveznici se u gornjem lijevom kutu nalazi popis svih programa pokretača s pripadajućim parametrima i njihovim objašnjenjem.

Nakon što je glavni program pokrenut, potrebno je pokrenuti program za čitanje ulaznih podataka te program za ispisivanje rezultata izvršavanja programa.

Program *Consumer* ispisuje rezultate izvođenja programa na standardni izlaz. Pokreće se idućom naredbom:

```
➢ mvn compile exec:java -D exec.mainClass=hr.mateo.kafka.Consumer
```

Kod 7.6 Naredba za pokretanje programa *Consumer*

Program *Producer* čita ulaznu *.csv* datoteku te šalje ulazne podatke glavnom programu. Pokreće se idućom naredbom:

```
➢ mvn compile exec:java -D exec.mainClass=hr.mateo.kafka.Producer
```

Kod 7.7 Naredba za pokretanje programa *Producer*

Kad glavni program završi s obradom podataka, *Consumer* će na standardni izlaz ispisati rezultat te dodatne parametre za analizu (vrijeme izvođenja programa, propusnost programa pokretača te propusnost čitavog sustava).

Ako korisnik želi veću kontrolu nad parametrima izvršavanja programa, postoji opcija da izravno pokrene izvršavanje programa koristeći odabrani alat za obradu toka podataka. Potrebno je pozicionirati se u direktorij gdje se nalazi datoteka *pom.xml* te izvršiti iduću naredbu:

```
➢ mvn package -Pspark-runner
```

Kod 7.8 Naredba za kreiranje *JAR* datoteke

Naredba će stvoriti novu *JAR* datoteku u direktoriju *apache-beam-test/target* pod nazivom *apache-beam-test-1.0-SNAPSHOT-shaded*.

Napomena: ako se nije generirala *JAR* datoteka, potrebno je opet pokrenuti naredbu.

Parametar koji dolazi nakon „*-P*“ osigurava kompatibilnost generirane *JAR* datoteke za odabrani program pokretač.

Generiranu *JAR* datoteku može se proslijediti alatu za obradu toka podataka te tako pokrenuti izvršavanje programa. Primjer naredbe koja pokreče obradu toka podataka koristeći alat Apache Spark ima idući oblik:

```
➢ spark-submit --class hr.mateo.beam.Beam --master local[24] target/apache-bea
  m-test-1.0-SNAPSHOT-shaded.jar --runner=SparkRunner --task=aggregation
```

Kod 7.9 Naredba za pokretanje izravnog izvršavanja programa koristeći alat Apache Spark

Prednost izravnog pokretanja izvršavanja programa je veća kontrola nad okolinom izvršavanja programa (memorija, broj jezgri, broj radnika itd.). Moguće je konfigurirati parametre u konfiguracijskim datotekama kako se ne bi morali prosljeđivati putem naredbenog retka prilikom svakog pokretanja programa.