Pokretanje aplikacije

Aplikacija je testirana na sledeće načine

- local
- Spark Standalone (pseudo distribuiran)
- Spark Standalone i HDFS (pseudo distribuiran)
- Spark Standalone i HDFS na docker kontejnerima, korišćenjem BDE docker slika

Priprema aplikacije

Svi fajlovi potrebni za pokretanje aplikacije se mogu kreirati pokretanjem skripte zip.sh koja se nalazi u repozitorijumu. Ova skripta kreira folder zip, u koji kopira main.py i kreira paket jobs.zip. Ovaj fajl i paket se mogu direktno koristiti za spark-submit.

Argumenti aplikacije

main.py --input input --output output --job {{ job }} [--debug]
--input: ulazni direktorijum, podržava sve fajl-sisteme koje podržava Spark file://, hdfs://, itd.
--output: izlazni direktorijum, podržava sve fajlsisteme koje podržava Spark file://, hdfs://, itd. U tom direktorijumu se kreira poddirektorijum sa imenom {{job_name}}_{{date}}, u kojem se nalaze podaci
--debug: prikazuje međukorake (za manji skup podataka)
--job:

- daily_statistics ili ds
- transportation_statistics ili ts
- ransportation_modes ili tm

Pokretanje u lokalnom modu

```
$SPARK_HOME/bin/spark-submit --py-files jobs.zip --master local[4] main.py --input /home/ana/input --
output /home/ana/output --job {{name}}
```

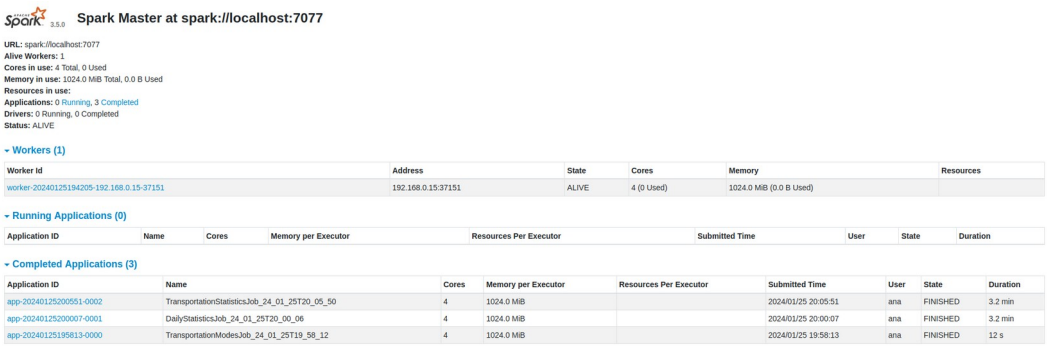
Pokretanje u Spark Standalone modu

Spark verzija 3.3.5. Dokumentacija.

Potrebno je pokrenuti spark master i spark worker. Spark start-all.sh skripta podrazumevano pokreće master i jedan worker na localhost-u. Nakon toga se aplikacija može proslediti korišćenjem spark-submit skripte.

```
$SPARK_HOME/sbin/start-all.sh
$SPARK_HOME/bin/spark-submit --py-files jobs.zip --master spark://localhost:7077 --executor-cores
{{cores}} --executor-memory {{memory}} main.py --input /home/ana/input --output /home/ana/output --job
{{name}}
```

Primer izvršenja aplikacije:



Spark Master at spark://localhost:7077

URL: spark://localhost:7077
Alive Workers: 1
Cores in use: 4 Total, 0 Used
Memory in use: 1024.0 MiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 3 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory	Resources
worker-20240125194205-192.168.0.15-37151	192.168.0.15:37151	ALIVE	4 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

Completed Applications (3)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20240125200551-0002	TransportationStatisticsJob_24_01_25T20_05_50	4	1024.0 MiB		2024/01/25 20:05:51	ana	FINISHED	3.2 min
app-20240125200007-0001	DailyStatisticsJob_24_01_25T20_00_06	4	1024.0 MiB		2024/01/25 20:00:07	ana	FINISHED	3.2 min
app-20240125195813-0000	TransportationModesJob_24_01_25T19_58_12	4	1024.0 MiB		2024/01/25 19:58:13	ana	FINISHED	12 s

Spark Standalone sa HDFS-om

1. Podešavanje HDFS-a u pseudo distribuiranom modu

Hadoop verzija 3.3.6 - Apache Docs

1. Podesiti java home
2. Instalirati ssh i pdsh, podesiti passphrasless ssh
3. Podesiti HDFS konfiguracione fajlove

```
$ export JAVA_HOME={{here}}
$ sudo apt-get install ssh
$ sudo apt-get install pdsh
$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
```

$HADOOP_HOME/etc/hadoop/core-site.xml:

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

$HADOOP_HOME/etc/hadoop/hdfs-site.xml:

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
```

$HADOOP_HOME/etc/hadoop.env.sh:

```
export PDSH_RCMD_TYPE=ssh
```

2. Pokretanje HDFS demona

Formatiratiranje namenode-a, pokretanje HDFS demona, kreiranje roditeljskog direktorijuma

```
$HADOOP_HOME/bin/hdfs namenode -format
$HADOOP_HOME/sbin/start-dfs.sh
$HADOOP_HOME/bin/hdfs dfs -mkdir -p /user/ana
```

Postavljanje podataka na HDFS:

```
$HADOOP_HOME/bin/hdfs dfs -put /home/ana/input input
```

3. Pokretanje Spark master-a

```
$SPARK_HOME/sbin/start-all.sh
```

4. Pokretanje aplikacije

```
$SPARK_HOME/bin/spark-submit --py-files jobs.zip --master spark://localhost:7077 --executor-cores
{{cores}} --executor-memory {{memory}} main.py --input hdfs://localhost:9000/user/ana/input --output
hdfs://localhost:9000/user/ana/output --job {{name}}
```

5. Dovlačenje podataka sa HDFS-a

```
$HADOOP_HOME/bin/hdfs dfs -get output /home/ana/output
```

6. Zaustavljanje Spark i HDFS daemona

```
$HADOOP_HOME/sbin/stop-dfs.sh
$SPARK_HOME/sbin/stop-all.sh
```

Pokretanje na klasteru Docker container-a

Korišćena BDE verzija: 3.1.2-hadoop3.2 (Dokumentacija)

Pokretanje klastera

Iz direktorijuma u kome se nalazi docker-compose fajl i .env fajl se može pokrenuti docker compose. Ovim se pokreće klaster docker kontejnera koji se sastoji od HDFS i spark demona.

```
$ docker network create bde --attachable
$ docker compose up -d
```

Podaci se kopiraju u namenode kontejner. Pokreće se interaktivni terminal na namenode kontejneru, gde se mogu izvršavati HDFS komande. Kreira se roditeljski direktorijum i u njega se postavljaju iskopirani podaci.

```
$ docker cp input namenode:/input
$ docker exec -it namenode bash
        # hdfs dfs -mkdir -p /user/ana
        # hdfs dfs -put input /user/ana/input
```

Pokretanje kontejnera koji će biti povezan na istu docker mrežu kao klaster i pozivanje spark submit iz tog kontejnera:

```
$ docker run -it --network bde --env-file hadoop.env -p 4040:4040 --name spark bde2020/spark-
base:3.1.2-hadoop3.2 bash

$ docker cp zip spark:/zip

# spark/bin/spark-submit --master spark://spark-master:7077 --py-files zip/jobs.zip
--executor-cores {{c}} --executor-memory {{m}} zip/main.py --input hdfs://namenode:9000/user/ana/input
--output hdfs://namenode:9000/user/ana/output --job {{name}}
```

Nakon uspešnog izvršenja, iskopirati podatak sa HDFSa na namenode, a zatim na host mašinu.

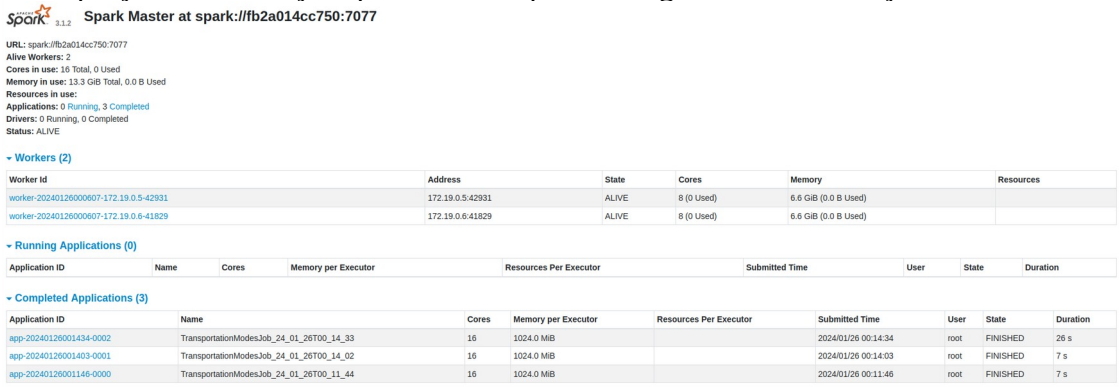
```
$ docker exec -it namenode bash
        # hdfs dfs -get /user/ana/output output
$ docker cp namenode:/output /home/ana/output
```

Primeri izvršenja

Lista kontejnera koji se izvršavaju:

```
ana@ana-PC:~/PycharmProjects/GeoLife$ docker ps
CONTAINER ID   IMAGE                                          COMMAND                  CREATED          STATUS
 PORTS                                                                                                  NAMES
265f9a5a79fa   bde2020/spark-base:3.1.2-hadoop3.2             "bash"                   18 seconds ago   Up 17 seconds
 0.0.0.0:4040->4040/tcp, :::4040->4040/tcp                                                              spark
1f18f5200a0d   bde2020/spark-worker:3.1.2-hadoop3.2           "/bin/bash /worker.sh"   2 minutes ago    Up 2 minutes
 8081/tcp, 0.0.0.0:8072->8071/tcp, :::8072->8071/tcp                                                    spark-worker-2
b2d821256979   bde2020/spark-worker:3.1.2-hadoop3.2           "/bin/bash /worker.sh"   2 minutes ago    Up 2 minutes
 0.0.0.0:8071->8071/tcp, :::8071->8071/tcp, 8081/tcp                                                    spark-worker-1
5752d3e5a3e5   bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   2 minutes ago    Up 2 minutes (healthy)
 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp, 0.0.0.0:9870->9870/tcp, :::9870->9870/tcp                   namenode
f907784dbe50   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   2 minutes ago    Up 2 minutes (healthy)
 9864/tcp                                                                                               datanode
fb2a014cc750   bde2020/spark-master:3.1.2-hadoop3.2           "/bin/bash /master.sh"   2 minutes ago    Up 2 minutes
 0.0.0.0:7077->7077/tcp, :::7077->7077/tcp, 6066/tcp, 8080/tcp, 0.0.0.0:8070->8070/tcp, :::8070->8070/tcp   spark-master
ana@ana-PC:~/PycharmProjects/GeoLife$
```

Pristupanje web interfejsu spark mastera pokrenutog na docker kontejneru:



Pristupanje web interfejsu namenode-a pokrenutog na docker kontejneru: