Big Data Management

Rengo Mattia
Cinti Alessandro
Rapone Ciro

Documento di Progetto

12 dicembre 2021 - MRColorR/supreme-pancake: Big Data Management project (github.com)

Panoramica

Per questo progetto è stato utilizzato il dataset facilmente consultabile a questo link, il quale contiene la misurazione della temperatura media, espressa in gradi Fahrenheit, delle principali città del mondo in un arco temporale che va dal 1995 al 2020. Prima di essere utilizzato, il dataset è stato leggermente modificato per una più semplice elaborazione. In particolare è stata rimossa la colonna “State” poiché vuota, è stato aggiunto un timestamp con ore-minuti-secondi casuali e infine sono stati rimossi i record della tabella dove la temperatura media presentava valori nulli. Dopo aver completato la pulizia del dataset si è passati alla stesura dell’intero progetto.

Obiettivo

L’ obiettivo di questo progetto è quello di raccogliere dati da sensori sparsi in tutto il globo per poi analizzarli. La raccolta di questi dati ha lo scopo di poter conoscere la temperatura giornaliera nelle città più importanti appartenenti ai vari continenti del mondo e visualizzarle per capire di quanto variano i valori col passare degli anni.

Specifiche

Per pulire il dataset è stato realizzato un semplice script in python. Per simulare la presenza dei sensori, i dati vengono inviati attraverso una piattaforma di eventi streaming chiamata Kafka e

successivamente attraverso il motore di esecuzione general purpose Spark questo flusso d'informazione viene elaborato e salvato nel column family database Cassandra. Infine, attraverso operazioni MapReduce, i dati vengono processati e caricati nel database. Utilizziamo il sistema distribuito HDFS composto da tre nodi con fattore di replicazione pari a 2 per memorizzare i dati dei sensori. In Cassandra è stato creato un KEYSPACE con fattore di replicazione anche qui posto a 2.

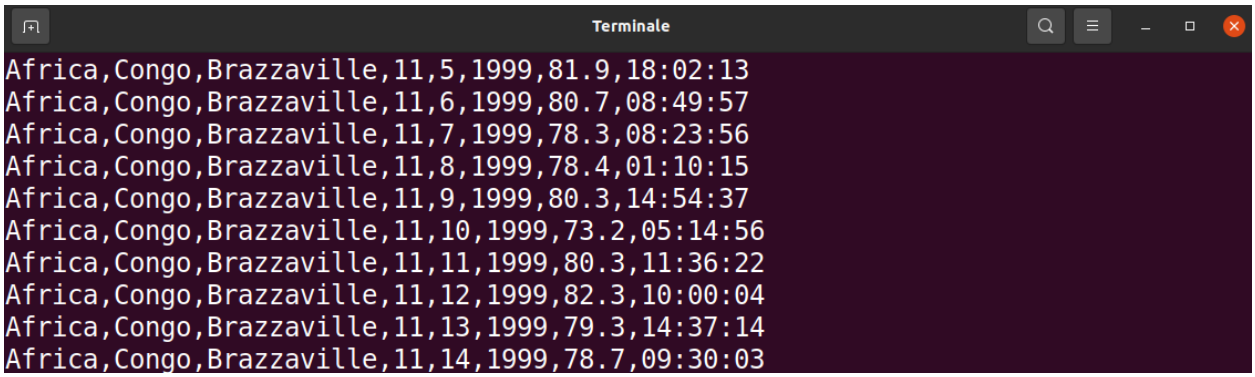
Il dataflow dell'applicazione è il seguente:

1. Dopo essere stato pulito, il dataset viene caricato in HDFS.
2. La classe KafkaProducer legge i dati da HDFS e li invia a Kafka simulando l'attività dei sensori.
3. La classe SparkProcessor periodicamente verifica se sono arrivati messaggi da Kafka, per poi elaborarli e caricarli in Cassandra.
4. La classe SparkProgram recupera i dati dal database e, attraverso operazioni MapReduce, calcola i valori della temperatura media per ogni anno e per ogni città (in una determinata regione appartenente al suo paese) e salva i risultati in apposite tabelle di Cassandra.

Esecuzione

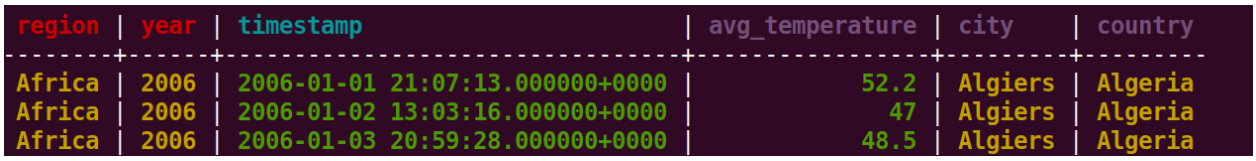
In prima battuta, è necessario eseguire lo script "start.sh" che avvierà Kafka, Hadoop e Cassandra. Se la cartella Kafka non è presente all'interno del progetto, è possibile scaricare la versione di Kafka 2.12-2.8.1 ed estrarla all'interno della root del progetto. Dopodiché, attraverso lo script "loadDatasetToHDFS.sh" il datatset viene pulito e inviato ad HDFS. Se si vuole far partire il programma scaricando direttamente il codice sorgente raccomandiamo di eseguire prima di tutto il comando "mvn clean package" (per eseguire la build del progetto). A questo punto è possibile eseguire lo script "run.sh", con il quale la classe KafkaProducer inizierà ad inviare i dati sul topic "*temperature*" e attraverso il comando qui sotto riportato è possibile osservare in tempo reale l'arrivo dei messaggi. Inoltre con questo script vengono eseguite operazioni real time con il motore MapReduce.

kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic temperature



Messaggi inviati nel topic temperature

A questo punto verrà eseguito lo SparkProcessor, il quale elabora i messaggi e li carica nella tabella “temperature” in Cassandra con KEYSPACE “iot”. Nel caso qui di seguito riportato è mostrato un esempio di record della tabella.



region	year	timestamp	avg_temperature	city	country
Africa	2006	2006-01-01 21:07:13.000000+0000	52.2	Algiers	Algeria
Africa	2006	2006-01-02 13:03:16.000000+0000	47	Algiers	Algeria
Africa	2006	2006-01-03 20:59:28.000000+0000	48.5	Algiers	Algeria

Una volta caricati i dati in Cassandra e aver eseguito le operazioni MapReduce, è possibile interrogare il database per ricavare i dati di interesse. Le operazioni eseguibili sono:

1. Ricerca della temperatura media per città in tutti gli anni;
2. Ricerca della temperatura media di ogni regione in tutti gli anni;
3. Ricerca della temperatura media annua globale;

Infine, per arrestare l’intero progetto è possibile eseguire lo script “stop.sh”.

Limiti e possibili estensioni

Per motivi di semplicità e di tempo si è scelto di inserire i parametri di configurazione all'interno del codice. Per favorire la portabilità, sarebbe opportuno inserire tutti i parametri in un file di testo facilmente accessibile (ad esempio un file con estensione ".properties"). A questo scopo sono stati resi parametrici gli host del cluster, permettendo la loro facile modifica e riconfigurazione attraverso il file "hosts" del sistema.