Andamento delle emissioni dei principali gas serra prodotti dall'uomo

Elena Burani, 327606

Indice

Elenco delle figure

1 Introduzione

L'obiettivo di questo progetto è lo sviluppo di un software in tecnologia Java per la gestione di un document database riguardante l'andamento delle emissioni dei principali gas serra di natura antropica. Viene utilizzata la teconologia Hadoop MapReduce con Input e Output da MongoDB con integrazione diretta.

1.1 Descrizione Dataset

Il dataset è stato progettato per aiutare a comprendere il contributo di specifici combustibili e settori per le emissioni di gas serra associate all'energia per ogni Paese a livello globale.
I dati presi in considerazione nel dataset si riferiscono alle emissioni totali di CO2 (emissioni derivanti dall'uso di energia e dai processi industriali, ad esempio la produzione di cemento), CH4 (emissioni di metano da rifiuti solidi, bestiame, estrazione di carbon fossile e lignite, risaie, agricoltura e perdite da gasdotti di gas naturale), protossido di azoto (N2O), idrofluorocarburi (HFC), perfluorocarburi (PFC) ed esafluoruro di zolfo (SF6). Nell'interpretare questi dati occorre tenere presente che si riferiscono alle emissioni dirette lorde, escludendo le emissioni o gli assorbimenti derivanti dal cambiamento di destinazione d'uso del suolo e dalla silvicoltura (LULUCF).

Il database è scaricabile dal sito https://datasource.kapsarc.org.


```
{
  "Year": 2015,
  "Country": "Italy",
  "Indicator": "Total  Emissions Excluding Lulucf",
  "Pollutant": "Methane",
  "Value": 43883.601
},
{
  "Year": 2015,
  "Country": "Hungary",
  "Indicator": "1 - Energy",
  "Pollutant": "Greenhouse Gases",
  "Value": 43118.632
},
{
  "Year": 2015,
  "Country": "Portugal",
  "Indicator": "Total  Emissions Excluding Lulucf",
  "Pollutant": "Hydrofluorocarbons",
  "Value": 2909.098
},
```


Figura 1: Dataset

Il database è formato da 2437 records, composto dai seguenti attributi:

- **Year**: specifica l'anno [Date];
- **Country**: nome della nazione [Text];
- **Sector o Indicator**: settore che sto andando a verificare [Text];
- **Pollutant**: tipo di emissione [Text];
- **Value**: specifica il valore delle emissioni [Decimal].

Il software sviluppato in questo lavoro prevede solo operazioni di interrogazione del database, che è di tipo read-intensive in quanto contiene informazioni utili prettamente a scopi di analisi. Si fonda sul paradigma MapReduce, tramite cui si possono svolgere operazioni di aggregazione fornendo una o due chiavi di aggregazione. Attraverso la sintassi <key=value> è possibile inserire una o più condizioni per filtrare i risultati.

2 Datflow e tecnologie utilizzate

2.1 Hadoop e MongoDB

Si è scelto di utilizzare un cluster Hadoop perchè è più performante del motore di esecuzione MongoDB ed inoltre un job fatto con Hadoop è più flessibile. Questo ci permette di leggere documenti che stanno in MongoDB, processarli con un job MapReduce ed esportare l'output in MongoDB. Una volta eseguito il programma, viene avviato il MongoClient che ci permette di accedere al database e alla collezione di input.
Inserendo il comando groupby vengono istanziate tutte le classi del framework MapReduce utili a questo tipo di operazione. La libreria utilizzata è mongo-hadoop-core, inclusa nel file .jar, necessaria per interfacciare il framework Hadoop con MongoDB.

2.1.1 Gasem

La classe Gasem, che in questo progetto svolge il ruolo di main invoca la classe Tools passando le impostazioni di set-up e i parametri utili per la configurazione dei job. Come formato di input viene impostato MongoInputFormat, come formato di output viene impostato MongoOutputFormat. Tramite il metodo set(String name, String value) della classe Tools si sceglie come trattare i parametri in formato String forniti dall'utente, cioè le condizioni di filtraggio (se specificate), la chiave di aggregazione, l'and o l'or logico da applicare (se presenti) e l'opzione di conteggio da utilizzare (se specificata). Questi parametri vengono poi passati ai metodi delle classi MapperClass e ReducerClass per permettere l'esecuzione dei map task e reduce task. All'interno della classe Tools vengono inoltre identificate le risorse di input e output con i metodi setInputURI e setOutputURI. Successivamente sono impostati i formati sia delle chiavi che dei valori delle coppie chiave-valore intermedie (in uscita dalla fase di map e in ingresso a quella di reduce) e di quelle in output. Le chiavi intermedie utilizzano i formati <Text, DoubleWritable>, invece quelle di output <BSONWritable, DoubleWritable>.

2.1.2 Funzione Map

All'interno della classe GasemMapper i problemi da gestire sono principalmente legati alla gestione dell'input fornito dall'utente, in quanto le sintassi fornite per le query possono generare molteplici combinazioni. Analizzando l'attributo Configuration della classe Context passata al metodo map è possibile estrarre la stringa contenente la query inserita dall'utente da interpretare. Inizialmente viene rilevata la tipologia di conteggio (conta normale, somma o valor medio), successivamente si gestisce il filtraggio dei dati in base alla presenza di una o più coppie chiave valore, distinguendo i casi and e or. Infine sono emesse le coppie chiave-valore intermedie utilizzando come chiave di aggregazione la chiave o la coppia di chiavi specificate in input, che verranno passate allo step di reduce.

2.1.3 Funzione Reduce

La Classe GasemReducer riceve in input le coppie chiave-valore già ordinate e partizionate, in seguito ad una fase di shuffle gestita dal framework in maniera trasparente. Il metodo reduce effettua il conteggio, la somma o la media dei valori della lista Iterable associata a ciascuna chiave ed emette le coppie chiave-valore generate come output.

2.1.4 Output

Al termine del processo di MapReduce il risultato dell'aggregazione viene salvato in un document database in formato .json in cui è riportato il numero di occorrenze relativo alla chiave di aggregazione. Il risultato sarà memorizzato in una collezione con un nome che la identifica univocamente.

3 Casi d'uso

Per poter utilizzare il software bisogna seguire i passaggi e le sintassi illustrati di seguito. Nella directory Gasem sono presenti:

- **S**cript bash per l'utilizzo del software;
- **U**na Cartella input nella quale sono presenti il dataset in formato json gasem.json ed il file jar mongodb-burani-jar-with-dependencies.jar contenente il codice sorgente;
- **U**na cartella output per i risultati delle query in formato json;
- **I**l file readme.txt contenente informazioni utili all'utilizzo del programma.

3.1 Script bash

In seguito verranno descritti gli script bash contenti le istruzioni di avvio, di inizializzazione e di arresto del sistema.

start.sh Per prima cosa è necessario eseguire lo script contenente i comandi di avvio di Hadoop e MongoDB.

```
#!/bin/bash

echo " MONGO-DB START-UP "
sudo service mongod start

echo " HADOOP START-UP "
$HADOOP_DIR/sbin/start-dfs.sh
$HADOOP_DIR/sbin/start-yarn.sh
jps

echo " COMPLETED"
```

Figura 2: Start

initialization.sh Successivamente viene avviato il database in MongoDB e vengono esportate le dipendenze necessarie al funzionamento.

```
#!/bin/bash

echo " DATABASE IMPORT"
mongoimport --db gasem --collection gasem --jsonArray --drop --file $PWD/input/gasem.json

echo " DEPENDENCIES EXPORT"
sh $HADOOP_DIR/etc/hadoop/hadoop-env.sh

echo " COMPLETED"
```

Figura 3: Inizialization

run.sh Lancia l'esecuzione del software che permette di svolgere le operazioni previste. In particolare, avvia l'esecuzione della classe Gasem del progetto.

```
#!/bin/bash

$HADOOP_DIR/bin/hadoop jar
$HADOOP_DIR/mongodb-burani-jar-with-dependencies.jar
bigdataman.burani.Gasem.Gasem gasem
```

Figura 4: run

Una volta eseguito questo script il sistema rimane in ascolto dell'input dell'utente.

stop.sh Contiene i comandi di arresto di Hadoop e MongoDB.

```
#!/bin/bash

echo " CLOSING MONGO-DB "
sudo service mongod stop

echo " CLOSING HADOOP "
$HADOOP_DIR/sbin/stop-dfs.sh
$HADOOP_DIR/sbin/stop-yarn.sh

echo " COMPLETED"
```

Figura 5: stop

3.2 Operazioni

In seguito all'esecuzione dello script run-mongodb.sh, il sistema rimane in ascolto di eventuali operazioni da svolgere. Quando vengono digitate le interrogazioni la relativa sintassi non può essere modificata.

Digitando il comando help, vengono visualizzate le operazioni che è possibile effettuare.

L'unico parametro strettamente necessario è la chiave di aggregazione o una coppia di chiavi di aggregazione.

Query con singola chiave di aggregazione e conteggio delle occorrenze.

Query con singola chiave di aggregazione e due clausole where con and logico.

Query con singola chiave di aggregazione e due clausole where con or logico.

Query con somma dell'Indicator.

```
{"_id": {"Country": "Australia"}, "value": 45.0}
{"_id": {"Country": "Austria"}, "value": 47.0}
{"_id": {"Country": "Belgium"}, "value": 45.0}
{"_id": {"Country": "Canada"}, "value": 50.0}
{"_id": {"Country": "Chile"}, "value": 39.0}
{"_id": {"Country": "Costa Rica"}, "value": 15.0}
{"_id": {"Country": "Czech Republic"}, "value": 44.0}
{"_id": {"Country": "Denmark"}, "value": 43.0}
{"_id": {"Country": "Estonia"}, "value": 40.0}
{"_id": {"Country": "European Union (28 Countries)"}, "value": 49.0}
{"_id": {"Country": "Finland"}, "value": 45.0}
{"_id": {"Country": "France"}, "value": 46.0}
{"_id": {"Country": "Germany"}, "value": 51.0}
{"_id": {"Country": "Greece"}, "value": 45.0}
{"_id": {"Country": "Hungary"}, "value": 44.0}
{"_id": {"Country": "Iceland"}, "value": 44.0}
{"_id": {"Country": "Indonesia"}, "value": 14.0}
{"_id": {"Country": "Ireland"}, "value": 45.0}
{"_id": {"Country": "Israel"}, "value": 32.0}
{"_id": {"Country": "Italy"}, "value": 49.0}
{"_id": {"Country": "Japan"}, "value": 46.0}
{"_id": {"Country": "Korea"}, "value": 41.0}
{"_id": {"Country": "Latvia"}, "value": 40.0}
{"_id": {"Country": "Lithuania"}, "value": 41.0}
{"_id": {"Country": "Luxembourg"}, "value": 42.0}
{"_id": {"Country": "Mexico"}, "value": 41.0}
{"_id": {"Country": "Netherlands"}, "value": 45.0}
{"_id": {"Country": "New Zealand"}, "value": 44.0}
{"_id": {"Country": "Norway"}, "value": 47.0}
{"_id": {"Country": "Oecd - Europe"}, "value": 55.0}
{"_id": {"Country": "Oecd - Total"}, "value": 55.0}
{"_id": {"Country": "Oecd America"}, "value": 30.0}
```


Figura 6: Groupby Country


```
{"_id": {"Indicator": "Total  Emissions Excluding Lulucf"}, "value": 1.0}
{"_id": {"Indicator": "Total GHG Excl. Lulucf, Index 1990=100"}, "value": 1.0}
{"_id": {"Indicator": "Total GHG Excl. Lulucf, Index 2000=100"}, "value": 1.0}
```


Figura 7: groupby Indicator Country=Canada Pollutant=Hydrofluorocarbons


```
{"_id": {"Pollutant": "Carbon Dioxide"}, "value": 6.0}
{"_id": {"Pollutant": "Greenhouse Gases"}, "value": 52.0}
{"_id": {"Pollutant": "Hydrofluorocarbons"}, "value": 6.0}
{"_id": {"Pollutant": "Methane"}, "value": 6.0}
{"_id": {"Pollutant": "Nitrogen Trifluoride"}, "value": 3.0}
{"_id": {"Pollutant": "Nitrous Oxide"}, "value": 6.0}
{"_id": {"Pollutant": "Perfluorocarbons"}, "value": 6.0}
{"_id": {"Pollutant": "Sulphur Hexafluoride"}, "value": 6.0}
{"_id": {"Pollutant": "Unspecified Mix of HFCS and PFCS"}, "value": 1.0}
```


Figura 8: groupby Pollutant or Country=France Country=Spain

Query con media dell'Indicaotr.

Le informazioni riguardanti le query effettuate possono essere visualizzate o

```
{"_id": {"Indicator": "1 - Energy"}, "value": 4.386913803599999E7}
{"_id": {"Indicator": "1a1 - Energy Industries"}, "value": 1.6802619926999997E7}
{"_id": {"Indicator": "1a2 - Manufacturing Industries and Construction"}, "value": 6214147.858}
{"_id": {"Indicator": "1a3 - Transport"}, "value": 1.1808033469000004E7}
{"_id": {"Indicator": "1a4 - Residential and Other Sectors"}, "value": 5826110.885000002}
{"_id": {"Indicator": "1a5 - Energy - Other"}, "value": 727633.332}
{"_id": {"Indicator": "1b - Fugitive Emissions From Fuels"}, "value": 1908826.4890000003}
{"_id": {"Indicator": "1c - Co2 From Transport and Storage"}, "value": 127.005}
{"_id": {"Indicator": "2- Industrial Processes and Product Use"}, "value": 4004139.4990000003}
{"_id": {"Indicator": "3 - Agriculture"}, "value": 5053071.129999999}
{"_id": {"Indicator": "5 - Waste"}, "value": 1613930.8659999995}
{"_id": {"Indicator": "6 - Other"}, "value": 46.458999999999996}
{"_id": {"Indicator": "Agriculture, Forestry and Other Land Use (Afolu)"}, "value": 1656311.0}
{"_id": {"Indicator": "Land Use, Land-Use Change and Forestry (Lulucf)"}, "value": -3683612.2579999994}
{"_id": {"Indicator": "Total  Emissions Excluding Lulucf"}, "value": 9.675973060300007E7}
{"_id": {"Indicator": "Total  Emissions Including Lulucf"}, "value": 4.796250845399999E7}
{"_id": {"Indicator": "Total GHG Excl. Lulucf Per Capita"}, "value": 443.7690000000001}
{"_id": {"Indicator": "Total GHG Excl. Lulucf Per Unit of GDP"}, "value": 12.792000000000002}
{"_id": {"Indicator": "Total GHG Excl. Lulucf, Index 1990=100"}, "value": 1.1023026368500005E8}
{"_id": {"Indicator": "Total GHG Excl. Lulucf, Index 2000=100"}, "value": 49964.96599999999}
```


Figura 9: Indicator sum


```
{"_id": {"Indicator": "1 - Energy"}, "value": 504242.96593103465}
{"_id": {"Indicator": "1a1 - Energy Industries"}, "value": 197677.8814941176}
{"_id": {"Indicator": "1a2 - Manufacturing Industries and Construction"}, "value": 73107.62185882354}
{"_id": {"Indicator": "1a3 - Transport"}, "value": 138918.04081176475}
{"_id": {"Indicator": "1a4 - Residential and Other Sectors"}, "value": 69358.46291666669}
{"_id": {"Indicator": "1a5 - Energy - Other"}, "value": 10860.198985074627}
{"_id": {"Indicator": "1b - Fugitive Emissions From Fuels"}, "value": 25116.13801315789}
{"_id": {"Indicator": "1c - Co2 From Transport and Storage"}, "value": 9.769615384615385}
{"_id": {"Indicator": "2- Industrial Processes and Product Use"}, "value": 46024.59194252873}
{"_id": {"Indicator": "3 - Agriculture"}, "value": 58081.27735632183}
{"_id": {"Indicator": "5 - Waste"}, "value": 18550.929494252876}
{"_id": {"Indicator": "6 - Other"}, "value": 3.8715833333333336}
{"_id": {"Indicator": "Agriculture, Forestry and Other Land Use (Afolu)"}, "value": 1656311.0}
{"_id": {"Indicator": "Land Use, Land-Use Change and Forestry (Lulucf)"}, "value": -85665.40134883723}
{"_id": {"Indicator": "Total  Emissions Excluding Lulucf"}, "value": 314154.9694902596}
{"_id": {"Indicator": "Total  Emissions Including Lulucf"}, "value": 1115407.1733488373}
{"_id": {"Indicator": "Total GHG Excl. Lulucf Per Capita"}, "value": 10.565928571428575}
{"_id": {"Indicator": "Total GHG Excl. Lulucf Per Unit of GDP"}, "value": 0.30457142857142855}
{"_id": {"Indicator": "Total GHG Excl. Lulucf, Index 1990=100"}, "value": 423962.55263461545}
{"_id": {"Indicator": "Total GHG Excl. Lulucf, Index 2000=100"}, "value": 170.5288941979523}
```


Figura 10: Indicator avg

tramite il comando log da shell oppure nel file log.txt all'interno della cartella output.

4 Limiti e possibili estensioni

E' stato notato che all'uscita della fase di reduce durante l'esecuzione del programma, il documento in output in formato .json non possiede una sintassi corretta per essere letto da un qualsiasi parser di documenti .json, in quanto ogni record in uscita dal viene emesso tra parentesi graffe, ma senza delle parentesi quadre che racchiudono tutti i record e le virgole di separazione tra un record e il successivo. Una possibile soluzione potrebbe essere l'implementazione di uno script che viene lanciato appena viene emesso il documento .json dopo la fase di reduce, che si occupi di effettuare le modifiche al file di testo necessarie per rendere la sintassi json corretta.

Importando il database in MongoDB ed eseguendo alcune operazioni è possibile eseguire anche operazioni di insert document, al contrario del software sviluppato in questo lavoro che prevede solo operazioni di interrogazione del database.