Final-Report

September 27, 2021

1 Relazione Finale

Gruppo - Free Data

Componenti Gruppo - Simone Garzarella

2 Indice

3 Introduzione

IMDb, acronimo di Internet Movie Database, è un sito di proprietà di Amazon che gestisce informazioni su film, attori, registi, personale di produzione, programmi televisivi e anche videogiochi.

Il dataset è composto da 5 file:

- title.akas.tsv
- title.basics.tsv
- title.principals.tsv
- title.ratings.tsv
- name.basics.tsv

Rispettivamente essi contengono:

- Elenco di film/serie tv con relative informazioni di carattere generale
- Elenco di film/serie tv con informazioni aggiuntive
- Informazioni sul cast del film o della serie tv
- Informazioni sul rating del film o della serie tv
- Informazioni aggiuntive sul cast del film o della serie tv

3.1 Descrizione Problema

Dopo una fase iniziale di analisi e processamento di dati si vogliono eseguire 4 job (descritti nel dettaglio più avanti) utilizzando Spark.

I file utilizzati sono stati caricati tutti su hdfs e i job sono stati eseguiti prendendo in input i dati dal file system distribuito.

Anche i risultati sono stati salvati tutti su hdfs.

3.2 Specifiche Hardware

I test sono stati eseguiti in locale e su cluster con macchine con queste caratteristiche: - **Locale:** Virtual Machine Ubuntu 21.04, CPU i7 , 6GB Ram e 100GB Hard Disk - **Cluster:** AWS EMR con 1 Master Node e 5 DataNode. Istanze m5.xlarge con 16GB RAM, 4 vCPU e 64GB di spazio.

4 Analisi Dataset

Di seguito vengono analizzati i cinque file del dataset per individuare eventuali preprocessamenti da effettuare. Inoltre viene anche descritto il processo per creare dataset più piccoli o grandi (con sampling) per effettuare i successivi test.

[6]:
```
import pandas as pd
```

4.1 Title Akas

I campi di questo dataset sono:

- `titleId` (string) – Identificatore alfanumerico univoco del titolo
- `ordering` (integer) – Un numero per identificare univocamente le righe per un determinato titolo
- `title` (string) – Il titolo del film
- `region` (string) – La regione per questa versione del film
- `language` (string) – La lingua del titolo
- `types` (array) – Insieme di attributi (enumerato) per questo titolo. Uno o più tra questi: “alternative”, “dvd”, “festival”, “tv”, “video”, “working”, “original”, “imdbDisplay”.
- `attributes` (array) – Termini aggiuntivi per descrivere il titolo (non enumerato).
- `isOriginalTitle` (boolean) – 0: titolo non originale; 1: titolo originale.

[ ]:
```
title_akas = pd.read_csv('dataset/title.akas.tsv', sep='\t', low_memory=False)
```

[ ]:
```
title_akas
```

```
[ ]: title_akas.nunique()
```

4.2 Title Basics

Il dataset con le informazioni sui titoli è così strutturato:

- `tconst` (string) – Identificatore alfanumerico univoco del titolo
- `titleType` (string) – Il formato/tipo del titolo (e.g. movie, short, tvseries, tvepisode, video, etc)
- `primaryTitle` (string) – Il titolo più utilizzato e famoso / il titolo utilizzato dai creatori del film al momento del rilascio sui prodotti promozionali
- `originalTitle` (string) – Il titolo originale nella lingua originale
- `isAdult` (boolean) - 0: titolo non per adulti; 1: titolo per adulti
- `startYear` (YYYY) – L'anno di pubblicazione del film / anno di inizio della serie tv
- `endYear` (YYYY) – Anno in cui è terminata la serie tv
- `runtimeMinutes` – Minutaggio del titolo
- `genres` (string array) – Include fino a tre generi pertinenti con quel titolo

```
[ ]: title_basics = pd.read_csv('dataset/title.basics.tsv', sep='\t',␣
     ↪low_memory=False)
```

```
[ ]: title_basics
```

```
[ ]: title_basics.nunique()
```

4.3 Title Principals

Il dataset con le informazioni sul cast è così strutturato:

- `tconst` (string) - Identificatore alfanumerico univoco del titolo
- `ordering` (integer) – Un numero per identificare univocamente le righe per un determinato titolo
- `nconst` (string) - Identificatore alfanumerico univoco per un nome/persona
- `category` (string) – La categoria del job in cui quella persona era coinvolta
- `job` (string) – Lo specifico job (se applicabile)
- `characters` (string) – Il nome del personaggio interpretato da quella persona (se applicabile)

```
[ ]: title_principals = pd.read_csv('dataset/title.principals.tsv', sep='\t',␣
     ↪low_memory=False)
```

```
[ ]: title_principals
```

```
[ ]: title_basics.nunique()
```

4.4 Title Ratings

Il dataset con le informazioni sul rating è così strutturato:

- `tconst` (string) - Identificatore alfanumerico univoco del titolo
- `averageRating` – Media pesata di tutte le valutazioni degli utenti

- `numVotes` – Numero di voti che quel titolo ha ricevuto

```
[ ]: title_ratings = pd.read_csv('dataset/title.ratings.tsv', sep='\t',␣
     ↪low_memory=False)
```

```
[ ]: title_ratings
```

```
[ ]: title_ratings.nunique()
```

4.5 Name Basics

Il dataset con le informazioni aggiuntive sul cast è così strutturato:

- `nconst` (string) - Identificatore alfanumerico univoco per un nome/persona
- `primaryName` (string) – Il nome con cui quella persona è maggiormente conosciuta
- `birthYear` (YYYY) – Anno di nascita
- `deathYear` (YYYY) - Anno di morte (se applicabile)
- `primaryProfession` (array of strings) – top 3 professioni di quella persona
- `knownForTitles` (array of tconsts) – titoli per cui quella persona è conosciuta

```
[ ]: name_basics = pd.read_csv('dataset/name.basics.tsv', sep='\t', low_memory=False)
```

```
[ ]: name_basics
```

```
[ ]: name_basics.nunique()
```

4.5.1 Creazione di dataset di varie dimensioni

Sono stati generati dataset di dimensioni (approssimativamente) di 256/512/1024MB, oltre al dataset originale che ha dimensioni ~1GB.

I file generati (con relativa dimensione precisa) hanno nome title_akas[size].tsv

- title_akas256.tsv (122.1MB)
- title_akas512.tsv (244.2MB)
- title_akas1024.tsv (488.3MB)
- title.akas.tsv (968.4MB)

La scelta dei record da includere è effettuata con un sampling randomico (con un seed preimpostato, per la ripetibiltà)

```
def sample_all_sizes(name, dataset):
    for size in dataset_sizes:
        n_rows = round(dataset.shape[0] * size)
        sampled_df = dataset.sample(n=n_rows, random_state=42, replace=True)
        filename =  name + '{}.tsv'.format(int(size*2048))
        sampled_df.to_csv(filename, index=False, sep='\t')
```

5 Job 1 - Analysis by Title

Deve generare un report contenente, per ciascuna titolo:

- il nome con cui il titolo è maggiormente conosciuto
- il numero di regioni in cui il titolo è stato pubblicato
- l'elenco di tali regioni
- il numero di lingue in cui il titolo è stato pubblicato
- l'elenco di tali lingue

Il report è ordinato per valori crescenti del `tconst`

Es. ((‘tt0018742’, ‘The Cameraman’), ((‘Regions: 9’, ‘FR,DK,SE,GR,IT,BG,FI,XWG,BR’), (‘Languages: 1’, ‘bg’)))

6 Job 2 - Analysis by Year

Generare un report contenente, per ciascun anno:

- il numero totale di titoli usciti in quell'anno
- per ogni tipo di opera, il numero totale di titoli usciti in quell'anno
- per ogni genere, il numero totale di titoli usciti in quell'anno
- il conteggio totale dei titoli per adulti e non, usciti in quell'anno

Il report è ordinato per valori crescenti dell'anno.

Es. (‘Year: 1874’, (((‘Total: 1’, ‘Types:’, {‘short’: 1}), (‘Genres:’, {‘Documentary’: 1, ‘Short’: 1})), (‘Is Adult:’, {‘0’: 1})))

7 Job 3 - Actors Ranking

Generare una classifica degli attori che appaiono in più titoli.

Per ogni attore viene visualizzato:

- il suo nome
- il numero di titoli in cui appare
- la sua/le sue professione/i
- i titoli per cui esso è maggiormente conosciuto

Es. (‘nm5744243’, ((‘Tina Dharamsey’, 2925, ‘production_designer’), ‘tt9025492,tt2801992,tt6978954,tt0435437’))

8 Job 4 - Rating Analysis

Genera un report contenente:

- per ogni anno e per ogni genere: la media dei rating dei titoli pubblicati in quell'anno appartenenti a quel genere

Es. (‘1927’, {‘Documentary’: 6.7, ‘Short’: 5.84, ‘Comedy’: 5.9, ‘Music’: 5.2, ‘Animation’: 5.15, ‘Adventure’: 5.0, ‘Drama’: 5.0, ‘Fantasy’: 7.1})

9 Risultati

Di seguito vengono illustrati i risultati ottenuti eseguendo i job, sia in locale che su cluster.

Inoltre vengono confrontati i tempi di esecuzione al variare delle dimensioni del dataset di input (descritte nella prima sezione).

9.1 Job 1 - Analysis by Title

Risultati del primo job

```
(('tt0018742', 'The Cameraman'), (('Regions: 9', 'FR,DK,SE,GR,IT,BG,FI,XWG,BR'), ('Languages: 1

(('tt0018749', 'The Cardboard Lover'), (('Regions: 3', 'TR,IT,SUHH'), ('Languages: 2', 'tr,ru')

(('tt0018756', 'Champagne'), (('Regions: 2', 'BG,AR'), ('Languages: 1', 'bg')))

(('tt0018773', 'The Circus'), (('Regions: 10', 'SUHH,CZ,PT,XWG,PL,CSHH,IT,SK,RO,UY'), ('Languag

(('tt0018836', 'The Divine Woman'), (('Regions: 7', 'US,SE,GR,AT,SUHH,PT,DK'), ('Languages: 1'

(('tt0018844', 'Don Diego i Pelageya'), (('Regions: 1', 'SUHH'), ('Languages: 1', 'ru')))

(('tt0018873', 'The Fall of the House of Usher'), (('Regions: 3', 'SUHH,GR,US'), ('Languages: 1

(('tt0018905', 'The Foreign Legion'), (('Regions: 2', 'TR,US'), ('Languages: 1', 'tr')))

(('tt0018927', 'The Garden of Eden'), (('Regions: 5', 'SE,SUHH,US,FI,AT'), ('Languages: 1', 'ru

(('tt0018940', 'Giuditta e Oloferne'), (('Regions: 2', 'IT,XWW'), ('Languages: 1', 'en')))

(('tt0018998', 'Couple on the Move'), (('Regions: 3', 'FR,ES,XWW'), ('Languages: 1', 'en')))

(('tt0019026', 'Dom v sugrobakh'), (('Regions: 2', 'SUHH,PL'), ('Languages: 1', 'ru')))

(('tt0019044', 'Just Married'), (('Regions: 3', 'TR,DE,SE'), ('Languages: 1', 'tr')))

(('tt0019071', 'The Last Command'), (('Regions: 10', 'DK,SUHH,SK,JP,AT,TR,SE,IT,PL,HU'), ('Lang

(('tt0019086', 'The Last Fort'), (('Regions: 1', 'XWW'), ('Languages: 1', 'en')))

(('tt0019098', 'Lilac Time'), (('Regions: 9', 'PT,FR,AT,GR,IE,DK,PL,GB,SE'), ('Languages: 1',

(('tt0019130', 'The Man Who Laughs'), (('Regions: 6', 'TR,DK,CA,SUHH,GR,US'), ('Languages: 3',

(('tt0019168', 'Miss Edith, Duchess'), (('Regions: 4', 'PT,FR,DK,XWW'), ('Languages: 1', 'en'))

...
```

Di seguito il grafico che confronta i tempi di esecuzione al variare della dimensione del dataset, sia

in cluster (linee rosse) che in locale (linee blu).

[ ]:
```
![Analysis by Title]("times/analysis_by_title.png")
```

9.2 Job 2 - Analysis by Year

Risultati del secondo job

```
('Year: 1874', ((('Total: 1', 'Types: ', {'short': 1}), ('Genres: ', {'Documentary': 1, 'Short

('Year: 1878', ((('Total: 1', 'Types: ', {'short': 1}), ('Genres: ', {'Documentary': 1, 'Short

('Year: 1881', ((('Total: 1', 'Types: ', {'short': 1}), ('Genres: ', {'Documentary': 1, 'Short

('Year: 1883', ((('Total: 1', 'Types: ', {'short': 1}), ('Genres: ', {'Documentary': 1, 'Short

('Year: 1885', ((('Total: 1', 'Types: ', {'short': 1}), ('Genres: ', {'Animation': 1, 'Short':

('Year: 1887', ((('Total: 45', 'Types: ', {'short': 45}), ('Genres: ', {'Short': 45, 'Sport': 1

('Year: 1888', ((('Total: 5', 'Types: ', {'short': 5}), ('Genres: ', {'Short': 5, 'Documentary

('Year: 1889', ((('Total: 2', 'Types: ', {'short': 2}), ('Genres: ', {'Documentary': 1, 'Short

('Year: 1890', ((('Total: 6', 'Types: ', {'short': 6}), ('Genres: ', {'Documentary': 2, 'Short

('Year: 1891', ((('Total: 10', 'Types: ', {'short': 10}), ('Genres: ', {'Short': 10, 'Action':

('Year: 1892', ((('Total: 9', 'Types: ', {'short': 9}), ('Genres: ', {'Animation': 3, 'Short':

('Year: 1893', ((('Total: 3', 'Types: ', {'short': 3}), ('Genres: ', {'Comedy': 1, 'Short': 3,

('Year: 1894', ((('Total: 97', 'Types: ', {'movie': 1, 'short': 96}), ('Genres: ', {'Romance':

...
```

Di seguito il grafico che confronta i tempi di esecuzione al variare della dimensione del dataset, sia in cluster (linee rosse) che in locale (linee blu).

[ ]:
```
![Analysis by Year]("times/analysis_by_year.png")
```

9.3 Job 3 - Actors Ranking

Risultati del terzo job

```
('nm0914844', (('Reg Watson', 5310, 'writer,producer,director'), 'tt0088580,tt0140761,tt007407

('nm0912726', (('Tony Warren', 3870, 'writer,actor'), 'tt1261038,tt0161131,tt0053494,tt0059175
```

```
('nm0318114', (('Johnny Gilbert', 3340, 'actor'), 'tt0105812,tt0098749,tt0083399,tt0159881'))

('nm0068589', (('William J. Bell', 3326, 'writer,producer,miscellaneous'), 'tt0092325,tt006965

('nm0871618', (('Alex Trebek', 3168, 'producer,actor'), 'tt0106179,tt0159881,tt0117723,tt01601

('nm5203198', (('Zama Habib', 3104, 'writer,producer,director'), 'tt3889862,tt2316500,tt799339

('nm0068347', (('Lee Phillip Bell', 2949, 'writer,producer,actress'), 'tt1539102,tt7293086,tt0

('nm5744243', (('Tina Dharamsey', 2925, 'production_designer'), 'tt9025492,tt2801992,tt6978954

('nm0554045', (('Henrique Martins', 2815, 'director,actor'), 'tt1567252,tt0209576,tt0344195,tt

('nm0001846', (('Vanna White', 2677, 'actress,producer'), 'tt0106761,tt0110622,tt0082467,tt008

('nm2276735', (('Sampurn Anand', 2619, 'writer,miscellaneous,director'), 'tt1889879,tt1888332,

('nm0001468', (('David Letterman', 2548, 'writer,producer,actor'), 'tt0106053,tt0083441,tt0119

('nm0836809', (('Adrián Suar', 2438, 'producer,writer,actor'), 'tt0205706,tt0204110,tt0456246,

('nm0739351', (('Carlos Romero', 2421, 'writer'), 'tt0358874,tt0243054,tt0215388,tt0211873'))

('nm0276899', (('Daniel Filho', 2359, 'director,producer,actor'), 'tt0154075,tt0287625,tt02897

('nm0022750', (('Paul Alter', 2344, 'director,producer,writer'), 'tt0264464,tt0068120,tt007106

('nm0163863', (('Dick Clark', 2303, 'producer,actor,miscellaneous'), 'tt0202179,tt0185049,tt00

...
```

Di seguito il grafico che confronta i tempi di esecuzione al variare della dimensione del dataset, sia in cluster (linee rosse) che in locale (linee blu).

[ ]:
```
![Actors Ranking]("times/actors_ranking.png")
```

9.4 Job 4 - Rating Analysis

Risultati del quarto job

```
('1874', {'Documentary': 7.0, 'Short': 7.0})

('1878', {'Documentary': 7.4, 'Short': 7.4})

('1881', {'Documentary': 5.3, 'Short': 5.3})

('1883', {'Documentary': 6.4, 'Short': 6.4})
```

```
('1885', {'Animation': 5.3, 'Short': 5.3})

('1887', {'Animation': 4.49, 'Short': 4.69, 'Documentary': 4.91, 'Sport': 5.0})

('1888', {'Documentary': 6.3, 'Short': 6.22})

('1889', {'Documentary': 5.3, 'Short': 5.4})

('1890', {'Short': 5.22, 'Documentary': 5.3})

('1891', {'Documentary': 4.9, 'Short': 4.93, 'Sport': 5.1, 'Action': 4.7})

('1892', {'Short': 5.23, 'Sport': 4.9, 'Animation': 6.27, 'Comedy': 6.5, 'Romance': 6.5, 'Docu

('1893', {'Documentary': 4.4, 'Short': 4.97, 'Comedy': 6.1})

('1894', {'Documentary': 5.04, 'Short': 4.91, 'Action': 5.23, 'Western': 5.1, 'Sport': 4.87, '

('1895', {'Documentary': 4.98, 'Short': 4.94, 'Drama': 4.7, 'Comedy': 5.4, 'News': 5.2, 'Sport

('1896', {'Documentary': 4.84, 'Short': 4.83, 'Fantasy': 4.9, 'Comedy': 4.73, 'Family': 5.3, '

...
```

Di seguito il grafico che confronta i tempi di esecuzione al variare della dimensione del dataset, sia in cluster (linee rosse) che in locale (linee blu).

[ ]:
```
![Rating Analysis]("times/rating_analysis.png")
```

10 Conclusioni

Per quanto riguarda la tecnologia utilizzata, Spark rimane uno standard di fatto in ambito big data, spiccando sulle altre tecnologie sia per l'efficienza, sia per la versatilità e la semplicità di utilizzo.

I tempi di esecuzione dei job risultano avere la stessa andatura in locale e su cluster, riscontrando una crescita di tipo esponenziale al variare della dimensione dell'input.

Su cluster si può apprezzare una diminuzione dei tempi rispetto al locale, che si nota di più per file di dimensioni maggiori o per job più onerosi. Infatti per file di dimensioni minori i tempi tra cluster e locale si equivalgono, a volte addirittura in locale le prestazioni sono migliori. Per file maggiori, soprattutto eseguendo job molto pesanti, su cluster si apprezza una notevole diminuzione dei tempi.

[ ]:
```
```