Gestão de Grandes Conjuntos de Dados

1º Trabalho Prático

-

Mestrado em Engenharia Informática
Universidade do Minho

Grupo nº 8	
PG41080	João Ribeiro Imperadeiro
PG41081	José Alberto Martins Boticas
PG41091	Nelson José Dias Teixeira
PG41851	Rui Miguel da Costa Meira

7 de Abril de 2020

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

Na primeira parte deste trabalho prático é requerida a concretização e avaliação experimental de tarefas de armazenamento e processamento de dados utilizando as ferramentas computacionais *Hadoop HDFS*, *HBase* e, ainda, o paradigma *MapReduce*. Por forma a realizar estas tarefas, os dados a utilizar para tal efeito correspondem ao conjunto de dados público do *IMDB*, que se encontram disponíveis em:

https://www.imdb.com/interfaces/

Ao longo deste documento vão também ser expostos todos os passos tomados durante a implementação das tarefas pedidas neste projeto, incluindo as decisões tomadas pelos elementos deste grupo a nível de algoritmos e parâmetros de configuração. Para além disso são ainda apresentadas todas as instruções que permitem executar e utilizar corretamente os programas desenvolvidos. Por fim, na fase final deste manuscrito, são exibidos os objetivos atingidos após a realização das tarefas propostas.

De salientar ainda que durante os capítulos que se seguem são identificadas algumas alternativas para concretizar as tarefas indicadas neste trabalho prático.

Capítulo 2

Implementação

Tal como foi enunciado anteriormente, neste projeto é globalmente solicitada a elaboração de duas tarefas. Apresentam-se de seguida as mesmas:

1. Carregar os dados do ficheiro *"title.basics.tsv.gz"* para uma tabela *HBase*;
2. Utilizando a tabela *HBase* do ponto acima e os restantes ficheiros presentes no *dataset* mencionado no capítulo anterior, computar os dados necessários para apresentar para cada ator uma página. Esta última deve conter:
   - nome, datas de nascimento e morte;
   - número total de filmes em que participou como ator;
   - títulos dos três filmes com melhor cotação em que participou.

   Estes dados devem ser armazenados numa tabela *HBase*.

Nas próximas secções são evidenciadas as implementações para cada uma destas tarefas bem como algumas sugestões alternativas que poderiam ser tomadas em consideração.

2.1 1ª Tarefa

Após descarregar o ficheiro *"title.basics.tsv.gz"* presente na hiperligação do capítulo anterior, os elementos que compõem este grupo optaram por converter o mesmo no formato *.tsv*. A tomada desta decisão deve-se ao facto deste último permitir a partição de dados (isto é, potencia o **paralelismo**), ao contrário do formato *.gz* (*gzip*), e, ainda, ser mais rápido e eficiente no processo de descompressão quando comparado com o formato *.bz2* (*bzip2*). Mostra-se na seguinte figura a instrução associada à descompressão do ficheiro *"title.basics.tsv.gz"*:

```
gzip -d title.basics.tsv.gz
```

Figura 2.1: 1ª Tarefa - Conversão do ficheiro *"title.basics.tsv.gz"* para o formato *.gz* (*gzip*)

Antes de observar os passos relativos à realização desta tarefa, passos esses que se encontram explicitamente indicados nos próximos subcapítulos, é importante salientar que a execução das soluções elaboradas nas secções 2.1.1 e 2.1.3 são efetuadas com recurso a um ficheiro denominado por *Dockerfile*. De forma a entender melhor a configuração do mesmo, revela-se a seguir o seu conteúdo:

```
FROM bde2020/hadoop-base
COPY target/TP1-1.0-SNAPSHOT.jar /
ENTRYPOINT ["hadoop", "jar", "/TP1-1.0-SNAPSHOT.jar", "ClassName"]
```

Figura 2.2: 1ª Tarefa - *Dockerfile*

Após esta observação, indica-se ainda as opções adotadas para a execução do ficheiro *Dockerfile* com o intuito de garantir uma execução válida das soluções implementadas:

```
--network docker-hbase_default
--env-file ../docker-hbase/hadoop.env
--env-file ../docker-hbase/hbase-distributed-local.env
```

Figura 2.3: 1ª Tarefa - *Dockerfile* - Opções de execução

2.1.1 Criação da tabela *HBase*

De forma a criar a tabela *HBase* intrínseca a esta tarefa, foi implementada uma classe *Java*, ***CreateTableMovies***, que, após conectar-se com a base de dados não relacional *HBase*, trata da sua criação e configuração. Durante esse processo, é produzida apenas uma família de colunas, intitulada por ***details***, onde será armazenada toda a informação associada aos dados do ficheiro *"title.basics.tsv.gz"*.

De notar também que atribuiu-se o nome ***movies*** à tabela gerada, tal como o nome da classe *Java* transparece.

Foi também criada uma classe *Java* adicional, ***DeleteTableMovies***, que trata de eliminar a tabela descrita anteriormente. Esta foi desenvolvida com o intuito de remover a tabela em causa caso esta deixe se ser necessária no futuro.

Apresenta-se de seguida o modelo da tabela *HBase* pretendido para a concretização desta tarefa:

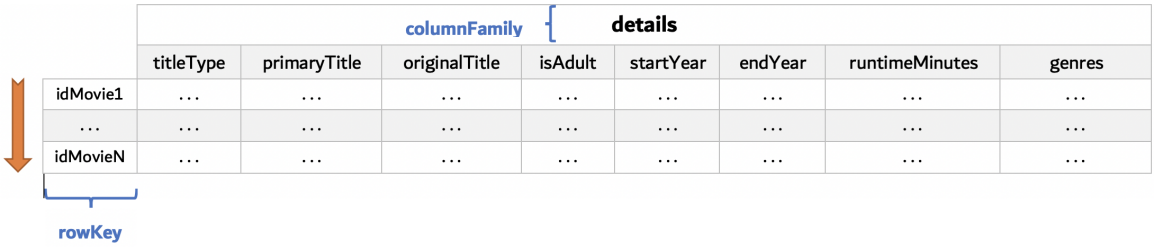

Figura 2.4: 1ª Tarefa - Modelo da tabela *HBase "movies"*

2.1.1.1 Alternativa

Uma possibilidade válida para realizar todo o processo associado à criação da tabela requerida seria utilizar a *HBase shell* de forma direta. Exibe-se de seguida as respetivas instruções:

```
docker run -it
           --network docker-hbase_default
           --env-file docker-hbase/hbase-distributed-local.env
           bde2020/hbase-base hbase shell
```

Figura 2.5: 1ª Tarefa : Alternativa - Acesso à *HBase shell*

```
hbase(main):001:0> create "movies", "details"
```

Figura 2.6: 1ª Tarefa : Alternativa - Criação da tabela *HBase "movies"*

Quanto à remoção da mesma tabela, à semelhança do procedimento tomado para a sua criação, adota-se a estratégia de usufruir explicitamente o mecanismo disponibilizado pela *HBase shell*:

```
hbase(main):001:0> disable "movies"
hbase(main):002:0> drop "movies"
```

Figura 2.7: 1ª Tarefa : Alternativa - Remoção da tabela *HBase "movies"*

2.1.2 Transferência do ficheiro para a plataforma *Hadoop HDFS*

De maneira a proceder ao carregamento do ficheiro *"title.basics.tsv"* para a plataforma *Hadoop HDFS* existem duas possibilidades. Antes de exibir estas últimas alternativas, foi criada uma pasta na plataforma *Hadoop HDFS*, denominada por *data*, onde serão colocados todos os ficheiros de *input* necessários. Exibe-se de seguida a instrução para tal efeito:

```
docker run --network docker-hbase_default
           --env-file docker-hbase/hadoop.env
           bde2020/hadoop-base hdfs dfs -mkdir /data
```

Figura 2.8: 1ª Tarefa - Criação da pasta `data` na plataforma *Hadoop HDFS*

Após a exposição deste comando, destacam-se nos próximos subcapítulos as duas alternativas mencionadas acima.

2.1.2.1 1ª Alternativa

Nesta possibilidade evidencia-se o campo `source` que corresponde à diretoria da pasta que contém o ficheiro *"title.basics.tsv"*. Dito isto, apresenta-se agora a primeira alternativa:

```
docker run --network docker-hbase_default
           --env-file docker-hbase/hadoop.env
           --mount type=bind,source="/path/to/local/folder/data",target=/data
           bde2020/hadoop-base hdfs dfs -put /data/title.basics.tsv /data
```

Figura 2.9: 1ª Tarefa : 1ª Alternativa - Transferência do ficheiro *"title.basics.tsv"* para a plataforma *Hadoop HDFS*

2.1.2.2 2ª Alternativa

Esta opção corresponde ao modo interativo de execução disponibilizado pela instrução *docker run*. Uma vez feita esta observação, expõe-se a seguir a segunda alternativa:

```
docker run -it
           --network docker-hbase_default
           --env-file docker-hbase/hadoop.env
           bde2020/hadoop-base bash

curl https://datasets.imdbws.com/title.basics.tsv.gz | gunzip |
hdfs dfs -put - hdfs://namenode:9000/data/title.basics.tsv
```

Figura 2.10: 1ª Tarefa : 2ª Alternativa - Transferência do ficheiro *"title.basics.tsv"* para a plataforma *Hadoop HDFS*

2.1.3 População da tabela *HBase*

Quanto à população da tabela criada previamente foi igualmente implementada uma classe *Java* para o efeito, designada por ***PopulateTableMovies***. Esta classe incorpora uma tarefa assente no paradigma *MapReduce*, onde é apenas elaborada a fase de *map*. Nessa mesma etapa é processada cada linha do ficheiro de *input* presente na plataforma *Hadoop HDFS* e, quando o tratamento estiver concluído, o resultado obtido é colocado na tabela *movies*.

2.2 2ª Tarefa

Tal como foi descrito no início do 2º capítulo deste documento, esta tarefa é composta por 3 alíneas distintas. Como tal é preciso tomar abordagens diferentes de forma a obter resultados corretos para cada uma das mesmas. Após uma leitura cuidadosa sobre o que é pedido em cada uma destas subtarefas, os elementos deste grupo notaram que iria ser necessário recolher 3 dos ficheiros que fazem parte do *dataset IMDB* para extrair os resultados pretendidos. Apresenta-se de seguida os 3 ficheiros escolhidos:

- *"title.basics.tsv"*: informação detalhada dos filmes, nomeadamente o ano de começo, géneros, entre outros;
- *"title.principals.tsv"*: conjunto de dados relativos aos atores que integram um determinado filme;
- *"title.ratings.tsv"*: informação associada à classificação e votação dos filmes presentes na plataforma *IMDB*.

Tal como se pode observar na listagem acima, o formato utilizado para estes ficheiros é o formato *.tsv*. A escolha desta extensão em detrimento de outras é exatamente a mesma que se encontra no início do subcapítulo relativo à 1ª tarefa.

2.2.1 Criação da tabela *HBase*

2.2.2 Transferência de ficheiros para a plataforma *Hadoop HDFS*

2.2.3 População da tabela *HBase*

2.2.3.1 Nome, datas de nascimento e morte do ator

2.2.3.2 Númeto total de filmes em que o ator participou

2.2.3.3 Títulos dos 3 filmes com melhores classificações em que o ator participou

Capítulo 3

Conclusão

Apêndice A

Observações

- Documentação *Java* 8:
  https://docs.oracle.com/javase/8/docs/api/
- *Maven*:
  https://maven.apache.org/
- *Hadoop*:
  https://hadoop.apache.org/
- *HBase*:
  https://hbase.apache.org/